2

WRDC-TR-89-2023

AD-A218 157

ETO - A TRAJECTORY PROGRAM FOR AEROSPACE VEHICLES

J. L. Leingang
W. A. Donaldson
K. A. Watson
L. R. Carreiro, 1LT

Analysis and Applications Branch
Advanced Propulsion Division

June 1989

Final Report for Period January 1987 to July 1988

Approved for public release, distribution unlimited.

AERO PROPULSION AND POWER LABORATORY
WRIGHT RESEARCH DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OHIO 45433-6563

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited Distribution Approved for Public Release
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) WRDC-TR-89-2023	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Aero Propulsion and Power Laboratory 6b. OFFICE SYMBOL (If applicable) WRDC/POPA	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB OH 45433-6563 Phone: (513) 255-4171	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61101F & 62204F	3012	08	93

11. TITLE (Include Security Classification) ETO - A TRAJECTORY PROGRAM FOR AEROSPACE VEHICLES

12. PERSONAL AUTHOR(S)
L. Leingang, W. A. Donaldson, K. A. Watson, L. R. Carreiro, 1Lt

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
FINAL	FROM Jan 87 TO Jul 88	June 1989	90

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Trajectory Program, Earth-to-Orbit, Cruise Vehicle Performance, Airbreathing Propulsion, Rocket Propulsion, Horizontal Takeoff, Single-Stage, Two-Stage Vehicles. (EG)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This is a formulation report and user guide for a two-degree-of-freedom flight trajectory computer program. The program, called ETO, runs on IBM/compatible XT/AT microcomputers. It is designed for very short run times, i.e., less than 1 to 3 minutes. Horizontal or low angle takeoff single-stage or two-stage vehicles can be studied. Ascent to earth orbit or constant speed cruise can be studied. Any combination of airbreathing or rocket propulsion may be used. Working from tabular data for propulsion performance, aerodynamics, and vehicle weight and fuel, the governing equations of motion are integrated forward in time to track velocity, altitude, range, flight path angle, and weight. A time ordered history of flight parameters is printed out. The program is available in Quick Basic and Fortran versions.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL John L. Leingang	22b. TELEPHONE NUMBER (Include Area Code) (513) 255-4171 22c. OFFICE SYMBOL WRDC/POPA

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

FOREWORD

This report describes the development and use of a two-degree-of-freedom flight vehicle trajectory program called ETO (Earth-to-Orbit). The program is intended primarily for study of the payload/range performance of airbreathing horizontal takeoff vehicles. The vehicles may be single stage or two stage and may be intended to reach earth orbit or to cruise at constant speed. Any combination of airbreathing propulsion and rocket propulsion may be used. Low angle ground launch or air launch vehicles may also be studied.

This report discusses the formulation of the computer program and gives detailed instructions on its use. Several example cases are included. The computer program is intended to be used on IBM or compatible -XT or -AT class microcomputers. The program is available on a 5¼-inch 360K byte floppy disk from the authors.

The flight vehicles used as examples in this report are not intended to represent any known or projected vehicle or level of technology. No inference should be made that the example vehicles typify current or projected capabilities.

This work was performed in the Analysis and Applications Branch, Advanced Propulsion Division of the WRDC (Wright Research Development Center) Aero Propulsion Laboratory. The work was done under in-house work unit 30120893. The work was begun in January 1987 and completed in July 1988.

The principal author was John L. Leingang. Co-authors were: Wayne A. Donaldson, Kenneth A. Watson, and 1Lt Louis R. Carreiro.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABBREVIATIONS AND SYMBOLS

Ac	Inlet cowl area, ft^2
Ao	Inlet air capture area, ft^2
Ao/Ac	Inlet air capture ratio, ft^2/ft^2
A Ref	Vehicle aerodynamic reference area, ft^2
C_D	Vehicle drag coefficient
C_{DO}	Vehicle zero lift drag coefficient
ΔC_D	Vehicle drag increment due to friction
C_L	Vehicle lift coefficient
$C_{L\alpha}$	Vehicle lift curve slope, 1/deg
D	Vehicle drag, lbf
dt	Integration step, secs
f/a	Airbreather fuel-air ratio
(f/a) STOICH	Airbreather stoichiometric fuel-air ratio
F_c	Vehicle centrifugal force relative to earth, lbf
F_n	Force component acting on vehicle normal to direction of motion, lbf
F_v	Force component acting on vehicle along axis of motion, lbf
g	Acceleration due to gravity, ft/sec^2
g_c	Acceleration constant, 32.174 $\frac{lbm/ft}{lbf/sec^2}$
h	Height, feet
h_o	Vehicle altitude, feet
Isp	Specific impulse, $\frac{lbf}{lbm/sec}$
Isp_A	Airbreather specific impulse, $\frac{lbf}{lbm/sec}$

ABBREVIATIONS AND SYMBOLS (Cont'd)

Isp_R	Rocket specific impulse, $\frac{lbf}{lbm/sec}$
K	Vehicle aerodynamic drag due to lift constant
L	Vehicle lift, lbf
M_e	Mass of the earth, 5.98×10^{27} grams
M_o	Flight Mach number
q	Atmospheric dynamic pressure, $1/2 \frac{\rho_o v_o^2}{g_c}$, lbf/ft^2
$q\ com_D$	Vehicle commanded dynamic pressure, lbf/ft^2
q_o max	Vehicle commanded maximum dynamic pressure, lbf/ft^2
q_o min	Vehicle commanded minimum dynamic pressure, lbf/ft^2
R	Vehicle flight range, nautical miles
R_e	Earth radius, 6.23×10^8 cm, 2.0899×10^7 feet
T	Vehicle thrust, lbf
t	Flight time, secs
v	Velocity, ft/sec
v_o	Vehicle flight velocity, ft/sec
v_n	Vehicle velocity normal to flight path, ft/sec
v_o temp	Vehicle velocity at which q_o is commanded to lower values to simulate a constant wall temperature flight path, ft/sec
W	Vehicle weight, lbm
$\dot{W}$	Vehicle rate of weight change, lbs/sec
$\dot{W}_{AIR}$	Airbreather propulsion air flow rate, lbm/sec
$\dot{W}_f$	Airbreather fuel flow, lbm/sec
$\dot{W}_p$	Rocket propellant flow, lbm/sec
α	Vehicle angle of attack, deg

ABBREVIATIONS AND SYMBOLS (Concluded)

γ	Vehicle flight path angle, deg
ϕ	Airbreather fuel equivalence ratio, (f/a)/(f/a) STOICH
ρ_0	Atmospheric density, lbm/ft^3

1.0 INTRODUCTION AND SUMMARY

A two-degree-of-freedom (2-DOF) digital computer program called ETO (Earth-to-Orbit) has been written. Five equations of motion describing weight, velocity, flight path angle, altitude, and range of a vehicle are integrated by a second-order routine. The program assumes a spherical, nonrotating earth. The primary purpose of the program is to evaluate horizontal takeoff earth-to-orbit vehicles of single- or two-stage configurations. Options are available in the program to evaluate cruise vehicles of various types, aircraft, and ground or air launched missiles.

Because the vehicle motion is in a single plane, i.e., 2-DOF, simplified aerodynamics are used in the form of user-input coefficients for zero-lift drag, the lift curve slope, drag due to lift factor, and friction drag. Airbreathing propulsion performance is user input as specific impulse vs fuel equivalence ratio and flight Mach number. These data along with input values of inlet air capture area are sufficient to define airbreathing thrust at every integration point. Rocket thrust is user input as specific impulse vs altitude and propellant flow rate vs Mach number. Airbreathing and rocket thrust may be used separately or simultaneously in any phase of the flight.

A number of options are available for tailoring earth-to-orbit missions or cruise missions. For earth-to-orbit missions, options exist to climb to orbit at a commanded constant dynamic pressure, along a varying dynamic pressure path, or a ballistic ascent beginning at a prescribed velocity. For cruise aircraft, constant altitude cruise or maximum L/D cruise may be specified.

For ground-launched missiles, launch at a prescribed initial angle is used. For air launches, an initial velocity and angle are prescribed. Missiles may be single stage or two stage.

A major objective of the program was to achieve reasonable accuracy and very short run times on microcomputers. Run times of 1 to 3 minutes were sought.

2.0 DEVELOPMENT AND DESCRIPTION OF PROGRAM ETO

2.1 Program Language and Compilers

Initially, the program was written in Microsoft GW-BASIC* with strict adherence to syntax that would permit it to run under the BASICA interpreter for debugging. Once a suitable version was obtained, it was compiled with the Microsoft BASIC Compiler, BASCOM. A little later in the program development, the Microway 87 BASIC compiler was obtained. It supported the Intel 8087/80287 floating point math chip, and program execution speed was improved greatly. Near the end of the code development cycle (a refinement phase), the Microsoft Quick BASIC 4.0 compiler and the Borland Turbo BASIC 1.0 compilers became available. They both support the 8087/80287 math chip, do not require line numbers in the codes, and support more structured programming. The code as listed in this report now conforms to the syntax requirements of the Microsoft Quick BASIC 4.0 compiler. It will not run under the BASIC interpreter. The integrated editing, debugging, and run environments of new Microsoft or Borland compilers greatly ease code development. They are available at prices as low as $65.

The source code is about 45,000 bytes, and a stand-alone executable file of the code is about 85,000 bytes. On -XT or -AT compatible machines with the 8087/80287 floating point math chip, very short run times are possible. The following gives some results obtained for the example problem TEST1.DAT, an earth-to-orbit flight:

Computer	Run Time, secs
Zenith Z248 (AT class)	
8 MHz 80286/5 MHz 80287	38 secs
8 MHz 80286/no math chip	210 secs
Sanyo 550 (Turbo XT class)	
7.2 MHz V-20 /8 MHz 8087	60 secs
ACER 1100	
16 MHz 80386/16 MHz 80387	12 secs
Vendex Turbo XT	
8 MHz 8088/no math chip	916 secs

* At time of publication, a Fortran version is now available. Contact authors for details.

2.2 Equations of Motion in a Plane

Figure 2-1* is a free body diagram of a flight vehicle in motion about a spherical, nonrotating earth. The vehicle, of mass W/g_c, is at a distance $(R_e + h_o)$ from the center of the earth of mass M_e. The vehicle is moving along a path at angle γ relative the earth. Vehicle thrust, T, acts along the axis of the vehicle. The vehicle is inclined to its axis of motion by its angle of attack, α. Lift, L, is exerted in a direction normal to the axis of motion, and drag, D, is exerted along the axis of motion. The weight, W, and the centrifugal force, F_c, act oppositely and along the radius to the earth.

The motion of the vehicle is described by its acceleration. It is most convenient to resolve the acceleration along the motion axis (the velocity axis) and along a direction normal to the motion axis. The acceleration along the motion axis will be defined from Newton's law for acceleration:

$$\frac{W}{g_c}\ \frac{dv}{dt} = \Sigma\ F_v$$

Resolving all of the forces to their components along the motion axis gives

$$\frac{W}{g_c}\ \frac{dv}{dt} = T \cos \alpha - D - W \frac{g}{g_c} \sin \gamma + F_c \sin \gamma$$

The acceleration normal to the axis of motion will be given by Newton's law for acceleration:

$$\frac{W}{g_c}\ \frac{dv_n}{dt} = \Sigma\ F_n$$

The rate of change of flight path angle is

$$\frac{d\gamma}{dt} = \frac{dv_n}{dt} \cdot \frac{d\gamma}{dv_n}$$

---

*Section 2 figures begin on page 11.

Since

$$\frac{d\gamma}{dv_n} = \frac{1}{v_o}$$

We have

$$\frac{d\gamma}{dt} = \frac{dv_n}{dt} \cdot \frac{1}{v_o}$$

and

$$\frac{dv_n}{dt} = v_o \frac{d\gamma}{dt}$$

So that Newton's law gives

$$\frac{W}{g_c} v_o \frac{d\gamma}{dt} = \Sigma F_n$$

The summation of normal forces gives

$$\frac{W}{g_c} v_o \frac{d\gamma}{dt} = L + T \sin \alpha - W \frac{g}{g_c} \cos \gamma + F_c \cos \gamma$$

The above two expressions, one for the axial acceleration and one for the normal acceleration, are sufficient to describe the time-dependent motion of the vehicle but are not yet sufficient to allow solution.

An expression for the time rate of change of weight is needed, and since drag and lift are also dependent on altitude, an expression for the time rate of change of altitude is needed. The time rate of change of range is not required but is of

interest for cruising flight. The final system of five equations to be integrated becomes

Weight change:

$$\frac{dW}{dt} = - \dot{W}$$

Velocity change:

$$\frac{dv_o}{dt} = \frac{g_c}{W}\left[T \cos \alpha - D - W \frac{g}{g_c} \sin \alpha + F_c \sin \gamma\right]$$

Flight path angle change:

$$\frac{d\gamma}{dt} = \frac{g_c}{v_o W}\left[L + T \sin \alpha + F_c \cos \gamma - W \frac{g}{g_c} \cos \gamma\right]$$

Altitude change:

$$\frac{dh}{dt} = v_o \sin \gamma$$

Range increment:

$$\frac{dR}{dt} = v_o \cos \gamma$$

The first four equations are coupled and must be integrated by employing an algorithm for integrating first-order systems of differential equations. As was stated above, the fifth equation, which tracks the range, is of interest for cruising flights. We decided to employ a simple two-step integration scheme (second-order Runge-Kutta, sometimes called Heun's method), and we hoped that its accuracy would be compatible with the choice of large time-integration steps, i.e., 2 to 5 seconds, and the desire to keep program execution time short.

The overall strategy for solving the problem of a flight trajectory is to integrate the coupled equations of motion forward in time, enforcing constraints/limits on values of the variables until some predefined value of one of the variables is reached and triggers a stop to the problem. The solution to the trajectory is the time-ordered history of values of all of the relevant variables up to the stopping condition.

At each integration time step, current values are needed for

$\dot{W}$ = rate of weight change (usually propellant flow)

W = weight

T = thrust

D = drag

L = lift

F_c = centrifugal force

α = vehicle angle of attack

γ = vehicle flight path angle

V_o = flight velocity

g = acceleration due to gravity (varies with altitude)

Constraints are applied by defining limits for such variables as α and γ. Thrust may be limited by commanded axial acceleration or by maximum fuel flow. Other constraints such as maximum and minimum flight dynamic pressure impose adjustments on α and thrust. Basically, a trajectory can be flown by using two commands, angle-of-attack and axial acceleration ("stick and throttle"). The trajectory proceeds through a series of flight phases (up to five are used in ETO) which

enforce the problem constraints. The main program module enforces the constraints appropriate to each flight phase. At the end of each integration step, the program returns to the beginning of this main module and "falls through" to the current flight phase to obtain the appropriate angle-of-attack and acceleration commands. The flight proceeds through phases based on computed events and on inputs from the vehicle input data file.

The flow diagram of Figure 2-2 shows the basic program strategy in which lift, drag, and the flight conditions (h_0, v_0) are defined; then the main module is entered to obtain angle-of-attack and acceleration commands, followed by generation of the current thrust value and finally going to the integration module. A runtime output step is taken, and the program returns to update lift, drag, and the flight condition (h_0, v_0). This process is repeated until the problem is completed.

The remaining parts of Section 2 give more specifics on how aerodynamics, propulsion, flight phases, and stopping conditions are handled.

2.3 Aerodynamics

Because the program is two-degree-of-freedom, i.e., motion in a plane, only trim aerodynamics are used. The term "trim aerodynamics" means use of lift and drag coefficient values at which the sum of all moments about the vehicle are zero. A data table of C_{DO} vs Mach number, a data table of $C_{L\alpha}$ and K vs Mach number and a table of friction increment to C_{DO} vs altitude defines the aerodynamics.

At a given Mach and altitude, the data tables are consulted for C_{DO}, $C_{L\alpha}$, K and ΔC_D then used in the following equations:

$$C_L = \alpha\, C_{L\alpha}$$

$$L = C_L\, q_0\, A\ Ref$$

$$C_D = C_{DO} + K\, C_L^2 + \Delta C_D$$

$$D = C_D\, q_0\, A\ Ref$$

2.4 Propulsion

Airbreathing propulsion is described by tables of Isp vs Mach number and equivalence ratio (ϕ), and a schedule of maximum ϕ vs Mach number. This information, along with tables of inlet capture area ratio (A_0/A_c) vs Mach and angle of attack, the inlet cowl area (A_c), and stoichiometric fuel-air ratio are sufficient to establish thrust, fuel flow, and airflow at each integration time step. At each time step, the following steps are taken to obtain thrust and fuel flow:

$$Ao = \left(\frac{A_0}{A_c}\right) A_c$$

$$\dot{W}_{AIR} = \rho_0 \; V_0 \; A_0$$

at a specified ϕ:

$$f/a = \phi \; (f/a) \; STOICH$$

$$\dot{W}_f = (f/a) \; \dot{W}_{AIR}$$

$$T = \dot{W}_f \; Isp$$

An interation scheme is used to adjust $\dot{W}_f$ by testing if T is greater or less than required to produce the commanded acceleration.

Rocket propulsion is described by tables of Isp vs altitude and by propellant flow rate ($\dot{W}_p$) vs Mach number:

$$T = \dot{W}_p \; Isp$$

Rocket thrust is modulated in response to the acceleration command. If the airbreather and the rocket are operating simultaneously, the rocket is throttled first to reduce thrust to the commanded value.

2.5 Flight Phases

Figure 2-3 shows a typical succession of flight phases as discussed in the following paragraphs.

Phase I is the takeoff roll. The vehicle accelerates to the takeoff/rotation velocity at zero lift and zero commanded climb. Once the rotation velocity is achieved, control passes to Phase II.

Phase II is a climb at a commanded pull-up load factor and axial acceleration. After the rotation velocity is exceeded in Phase I, the vehicle is permitted to pitch up to a maximum angle of attack or a commanded load factor, whichever limit is controlling. The vehicle climbs and accelerates to a commanded maximum subsonic velocity which cannot be exceeded until a specified altitude is reached. Once this Mach and altitude threshold is reached, angle of attack is commanded to zero, and the vehicle accelerates on a ballistic path (no induced drag) through the transonic region until a commanded dynamic pressure is reached. This completes Phase II.

Phase III is a climb and acceleration at a commanded dynamic pressure, q com. This value of q may be constant or may be made linearly variable with velocity, beginning at a specified velocity. A maximum and a minimum value of q is specified. A control algorithm generates angle-of-attack and acceleration commands at each integration step to keep the vehicle within the minimum and maximum q limits. The acceleration command is generated by examining whether q is greater or less than the commanded value and whether, in either case, it is getting larger or smaller. The algorithm for implementing this control is shown in Figure 2-4. A more detailed discussion is in Section 3.4. The output of this algorithm is a new value of α, angle of attack, to be used in the next integration step. With proper choices for the adjustment coefficients, a relatively smooth ascent path can be achieved. Once the α command is generated, an acceleration command is generated. The algorithm for this can be visualized from Figure 2-5. Basically, the strategy is that if q is greater than the maximum, then no acceleration is permitted. If q is less than the commanded value, then the maximum available acceleration is commanded. In the region where q is greater than commanded but less than maximum, a proportional reduction in acceleration is commanded as suggested by Figure 2-5. Acceleration along the commanded q path continues until a cruise command or a ballistic path command is made.

Phase IV generates a commanded zero acceleration to maintain a specified velocity at a commanded q (altitude) or to maintain velocity at maximum L/D. Maximum L/D cruise

is accomplished by finding the angle of attack that produces maximum L/D and then maintaining it to fuel exhaustion.

Phase V is a ballistic ascent and is normally commanded after having proceeded along a Phase III commanded q path. It is used primarily for earth-to-orbit flights. At a commanded velocity, the vehicle pulls up to a specified angle of attack, maintaining it until a specified flight path angle is achieved. Once the flight path angle is reached, the angle of attack is commanded to zero and the flight is ballistic, i.e., angle of attack equals zero from this point on. During Phase V an acceleration command different from that applied in Phase III may be used.

2.6 Stopping Conditions

Three stopping conditions are utilized: fuel exhaustion, achievement of commanded velocity, and achievement of the orbital condition. When the vehicle weight falls to the specified final value, the program stops and issues a message to that effect. For two-stage vehicles, the first stage is usually flown to a prescribed velocity or to fuel exhaustion. When either occurs, the program halts, issues a message that the first-stage flight has been completed and asks if you care to continue the problem with the second stage.

The orbital condition is defined by the centrifugal force becoming equal to the weight. When this occurs the program stops and advises that orbit has been achieved. Because the code is assuming a nonrotating earth, the orbital velocity required is always equivalent to be that of a polar orbit. Normally this will be slightly under 25,900 fps. The exact value depends on altitude. An Eastern equatorial orbit would be the minimum required orbital velocity and would be on the order of 24,500 fps. The user could specify a final velocity less than the polar orbit value that is appropriate to the orbit inclination of interest to him.

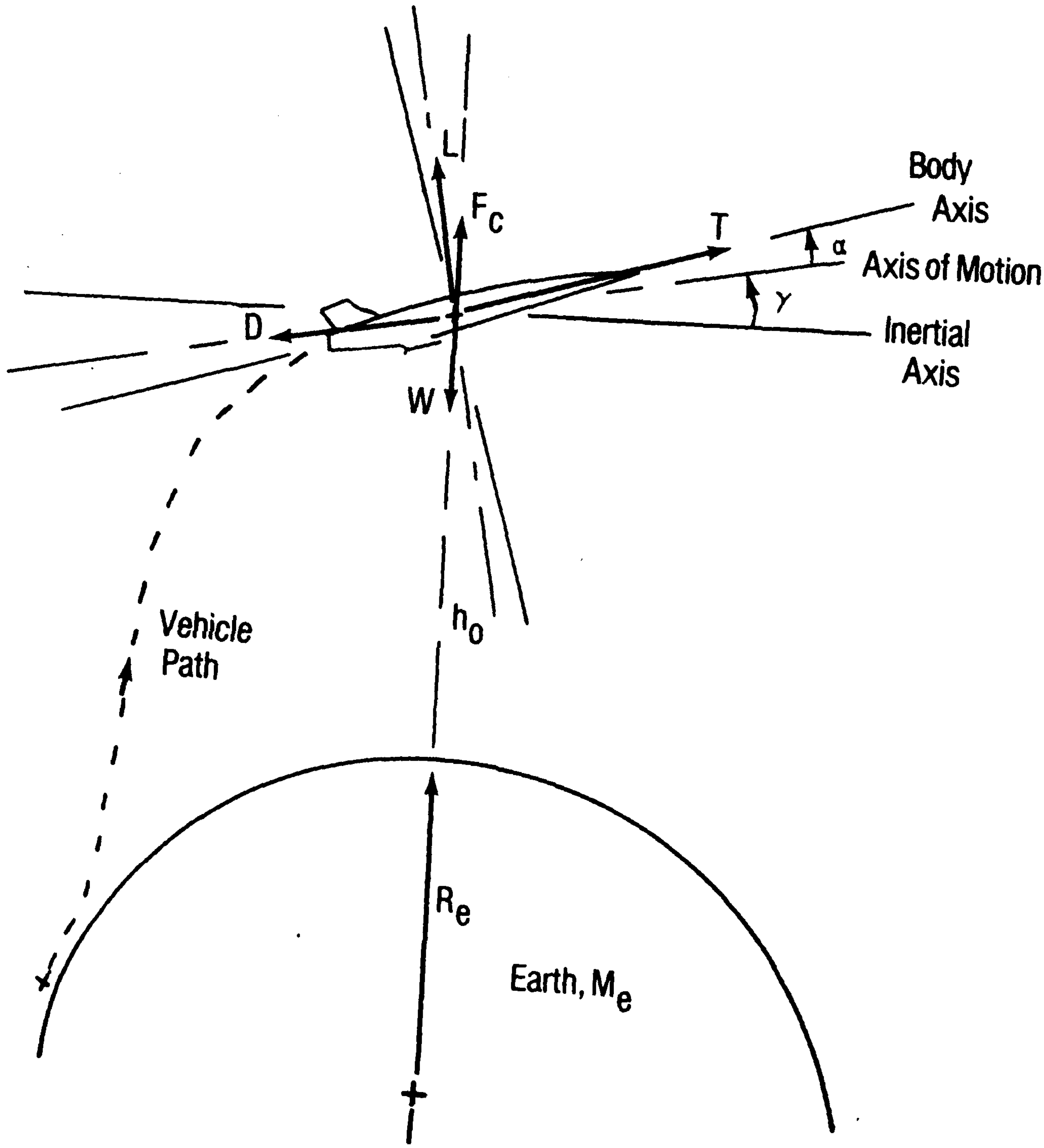


Figure 2-1. Free Body Diagram of Flight Vehicle

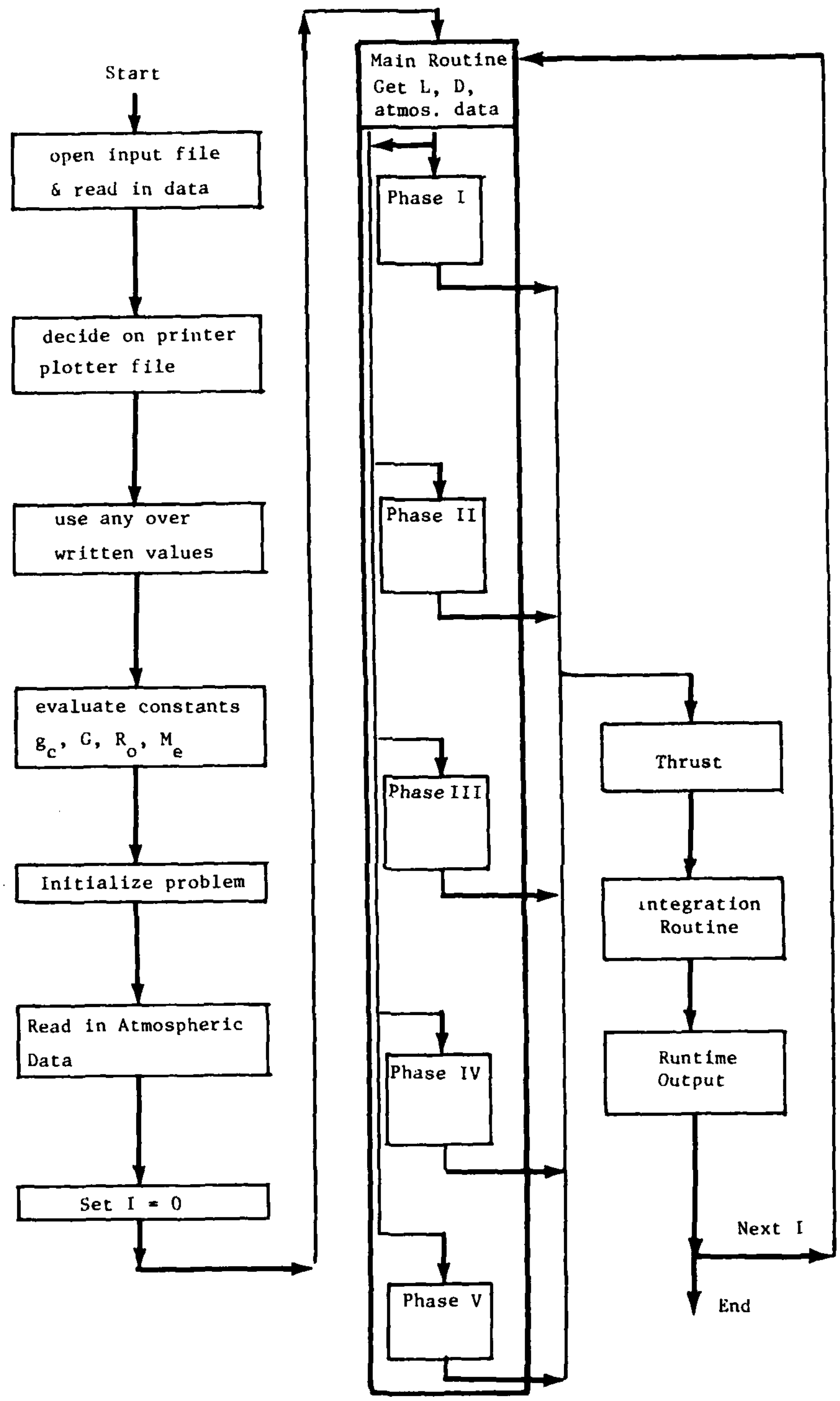


Figure 2-2. Program ETO Block Diagram

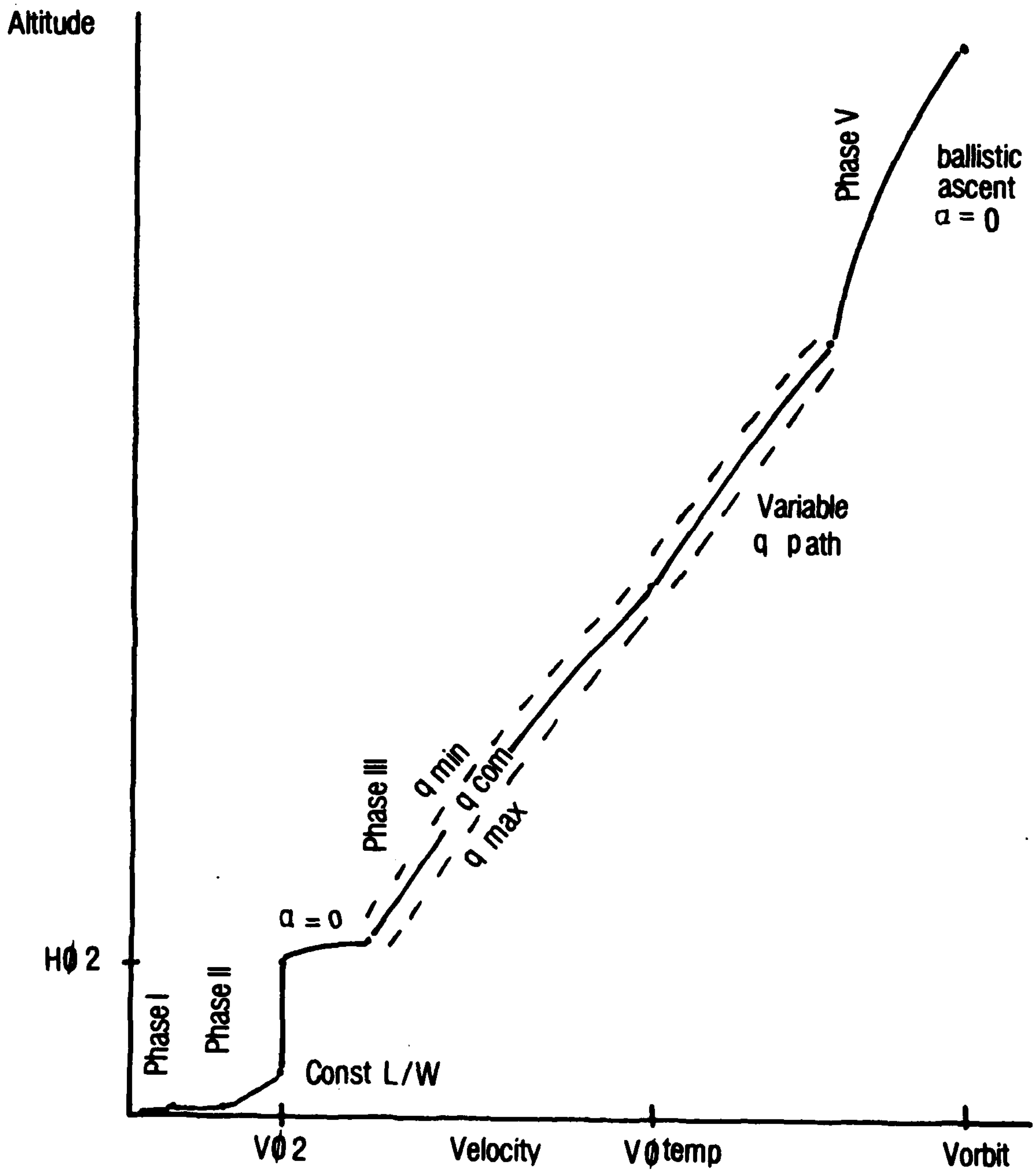


Figure 2-3. Vehicle Flight Trajectory

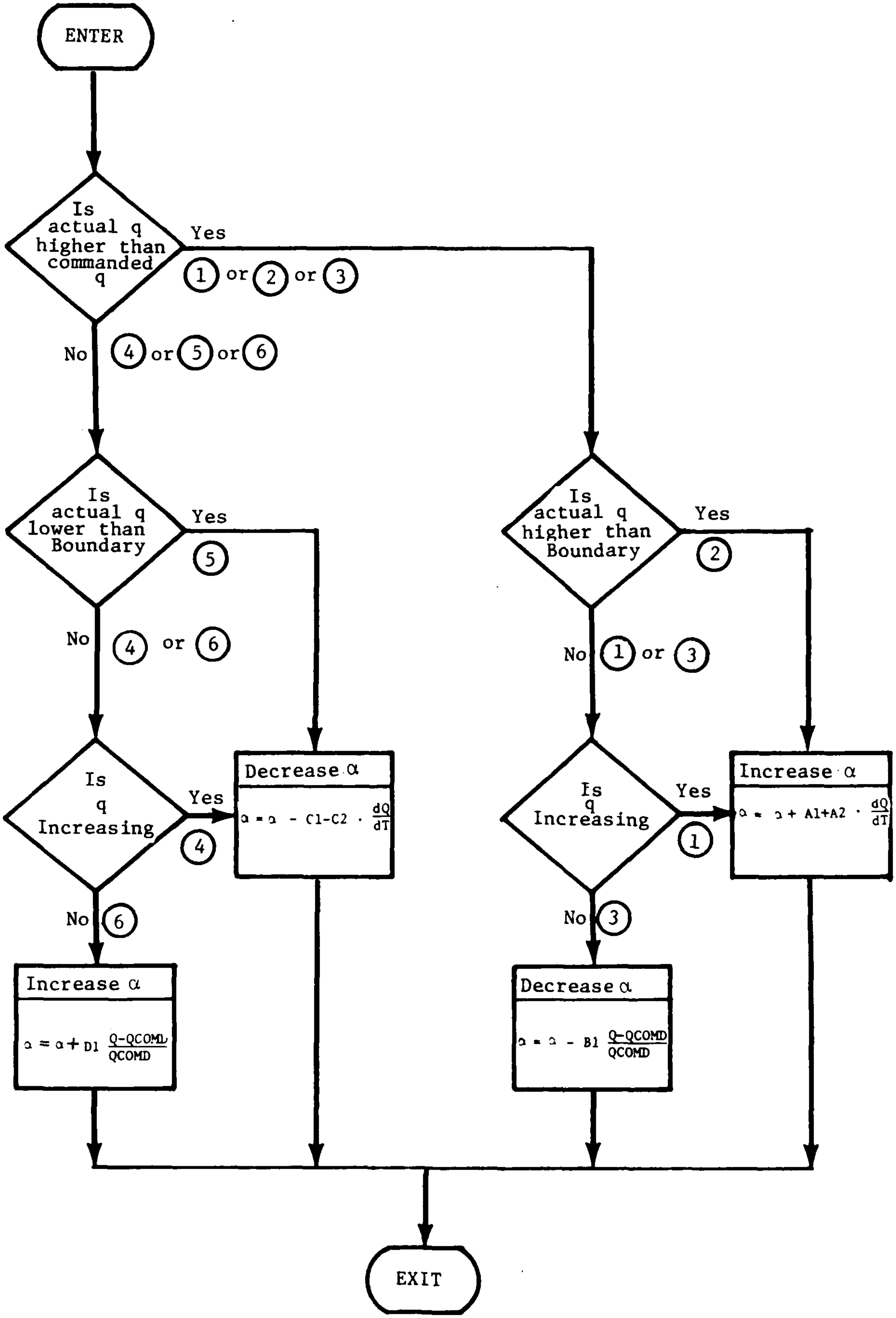


Figure 2-4. Angle-of-Attack Command

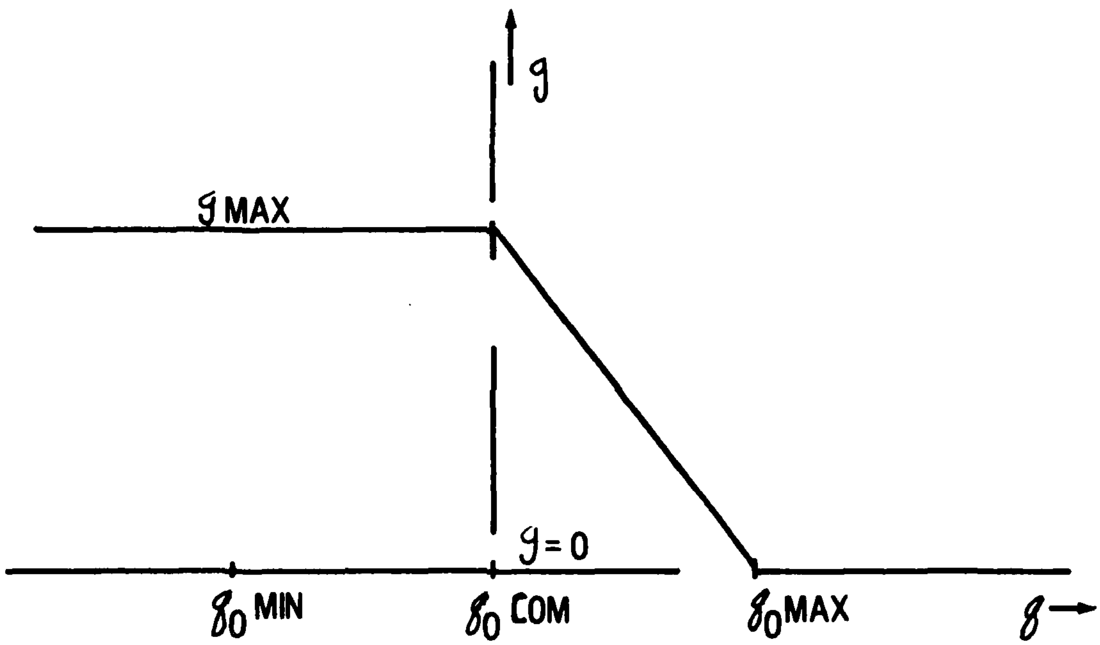


Figure 2-5. Acceleration Command

3.0 RUNNING PROGRAM ETO

3.1 Hardware Requirements

The distribution disk is a 360K IBM/MS-DOS format 5 ¼-inch floppy disk which contains the following files:

ET013.BAS (See Note Below)
ET013.BAC
ET013.EXE
TEST1.DAT
STG1.DAT
STG2.DAT
STG2BAL.DAT
ETO.TAB
GUID14.BAS
ET0123.BAS
ET0123.EXE

ET013.BAS is the program source code. ET013.BAC is a backup copy of the source code and ET013.EXE is a stand-alone executable version of the program. The designation ET013 reflects the version number of program ETO available at the time of this report. Significant upgrades will have higher numbers. The files with a .DAT suffix are sample input data files. Each data file illustrates a significant feature of the program. Each of these .DAT files is discussed in Section 3.6. The file ETO.TAB is an alphabetical listing with description of all the program variables. The file GUID14.BAS is an optional angle-of-attack module for flight Phase III which the user may substitute for GUID13 in the trajectory program (Section 2.5). ET0123.BAS and ET0123.EXE are the source code and executable file, respectively, which may be used to generate a file readable by LOTUS 123 of the program output file. This is included for users who may wish to use LOTUS 123 to generate plots of the program output.

NOTE: Since completion of this report FORTRAN versions are also available. Contact the authors for more information.

The minimum hardware equipment for running the stand-alone .EXE version of ETO is an IBM or compatible XT or AT microcomputer using MS.DOS 2.1 or higher. DOS versions earlier than 2.1 will not work properly. One hundred percent IBM compatibility does not appear to be required, however. A single disk drive (360K, 5¼-inch floppy disk) and 256K RAM are needed. It may be possible to run the program with less than 256K RAM but this has not been demonstrated. A math coprocessor, 8087/80287, is highly desirable for reducing run times but is not required. A printer is required for printed output of the time history of a flight. An Epson FX or compatible dot matrix printer is the best choice. With this printer letter-width paper can be used because this printer accepts the program commands to set compressed print. If your letter-width printer cannot respond to the program command for compressed print and cannot be set by a hardware switch, you will need to employ a wide-carriage printer.

The most convenient environment for running the program is from within the Microsoft Quick BASIC 4.0 editor/compiler. The computer will have to be 100 percent IBM compatible. The program was developed using this compiler, and the stand-alone .EXE version was generated by it. MS-DOS versions 2.11, 3.1, and 3.2 were used in the development. It does require more computer hardware resources, however. Microsoft advises that 320K RAM is needed. The minimum RAM used during development was 512K. It was satisfactory, but it is not known how close this is to the minimum needed for this program. Two 360K floppy disk drives are needed. A hard drive is much more convenient and permits quicker compilation. A RAM drive is also convenient. Quick BASIC 4.0 senses if a math coprocessor (8087/80287) is present and uses it if available. Version 4.0 of Quick BASIC is distinctly different from earlier versions. These earlier versions are not recommended.

Some limited development work was done using the Borland Turbo BASIC 1.0 editor/compiler. It does not require 100 percent IBM compatibility. It provides an integrated program development and compiler environment. Its choice over the Microsoft product is a matter of personal preference. To use the Borland compiler, the program variable AT would have to be changed to some other name. AT is a reserved work in Turbo BASIC. Some work would be needed in formatting the screen and printer output. Either the Microsoft or the Borland compiler will be required if the user desires to modify the code and recompile it.

3.2 Using the Stand-Alone .EXE Version of ETO

The first step should be to make a backup copy of the distribution disk. The backup copy should be used for running the program. We recommended that you copy your version of the MS-DOS COMMAND.COM file to the backup disk. The reason for this is that the program uses the BASIC "SHELL" command to temporarily exit to DOS to copy the input data file to the printer when the program prints out results. The program needs access to COMMAND.COM on the default drive in order for the "SHELL" command to work properly. MS-DOS 2.1 or greater is required.

Once the above has been accomplished, the program is ready to run. At the DOS prompt, type ET013 and press RETURN. The screen clears, and the following will be displayed:

IS THIS A SINGLE-STAGE (1), OR A TWO-STAGE VEHICLE (2)?

Respond by entering a 1 or a 2 and pressing the ENTER key. If you responded to the above with a 1, you will be prompted with:

SINGLE-STAGE FILE NAME?

Enter the name of the input file following the DOS file name convention, DRIVE: NAME.EXT. The input file does not need to be on the default drive.

If you had responded with a 2 to the first question, you would be asked the following:

FIRST-STAGE FILE NAME?

Again enter input file name as DRIVE: NAME.EXT and press RETURN key.

You will then be prompted with:

SECOND-STAGE FILE NAME?

Respond as above.

Next you will be asked:

"Do you want a listing of tabular input data (Y/N)?"

Normally you will not want this and will respond with a "no" (N).

If you respond with "yes" (Y), then the nine data tables in the first/single-stage input file are printed out. The printout is convenient for examining the data tables for accuracy.

The next prompt will be:

"Do you want printer output during run (Y/N)?"

As above, you will usually not want this. The printer output format is not particularly convenient and the program runs much more slowly in this mode. The principal use for this printout is diagnosis of problems early in the flight. The program can be stopped after the desired flight time has passed. The printer output interval is the same as the screen print interval, DELPRINT.

Whether you respond with "Y" or "N" the next screen message displays the value of the last item in the input file. If the screen shows a value different from what you intended, then the input file is in error. Also you will be prompted with the following:

"Do you want a plotter/printer file created (Y/N)?"

If you respond with "Y", then you will be prompted with:

"Enter the plotter/printer file name (DRIVE: NAME.EXT)".

After entering the desired drive and file name, press RETURN and the following will appear:

```
At the prompt below,enter a value for FREQ,the interval of plotter/
printer output.
A value of 1 for FREQ means output at each time increment
A value of 2 means every other time increment,etc. As a guide to
the amount of disk space required,use the following:

 BYTES REQUIRED = (200) X  (FLIGHT TIME)
                  ---------------------------------
                  (INTEGRATION INTERVAL) X (FREQ)

 Very large files exceeding 200K may be generated.Exit the
 program by pressing CTRL C if disk space is insufficient,
 otherwise enter value for FREQ below:
```

In response, enter a value that is some integer multiple of the integration interval, e.g., a FREQ value of 20 means that output will be written to the disk every 20 integration intervals. If the integration interval is 5 seconds, then the disk file would contain the flight history in 100-second increments. After entering the desired value for FREQ and pressing ENTER, the screen clears and the problem solution begins. As the solution proceeds, the screen will display the following type data:

```
T=2000.0  PHASE 3,CLIMB AT COMMANDED Q************
V0=15628.3 M0=15.1  ALPHA=  1.99          WDOTF=  20.05    RES1=    0.0
H0=129013.5         GAMMA=  0.10          WDOTA= 431.83    WDOTP=   0.0
 R=  2304.32           Q0=1045.90           PHI= 1.590
ISP= 1357.00     ALPHAMAX=  5.00             WT=  71861.9   L/W =  0.544
ISPEFF= 956.7    ALPHAMIN= -5.00             FC=  25796.4
ISPMEAN=1431.8    ACCCOM =  0.27           LIFT=  39107.9   L/D =  4.967
ISPAVG= 2642.7   ACCEL-G'S=  0.27          DRAG=   7873.4
ISPA/B= 1357.0                           THRUST=  27209.2   T/D =  3.456
ISPROC=    0.0                        THRUSTCOM=  27209.2
                                      THRUSTMAX=  42899.2   T/W =  0.379
```

The screen is updated at the frequency DELPRINT, a user input to the input data file. A DELPRINT value of 20 to 40 will insure that the speed of the solution is not limited by writing to the screen. The user should experiment with values of DELPRINT to find the values that are frequent enough to track the calculation progress without impeding the solution speed. The plotter/printer file that is generated has values of the following 25 variables recorded every FREQ X DT seconds of the flight:

Variable #	Variable	Units
1	TIMEX	secs
2	VØ	ft/sec
3	HØ	feet
4	RANGE/6Ø76	nautical miles
5	ISP	secs
6	ISPEFF	secs
7	ISPMEAN	secs
8	WDOTF	lbs/sec
9	WDOTA	lbs/sec
10	PHI	fraction of stoichiometric
11	THRUST	lbs
12	THRUSTCOM	lbs
13	THRUSTMAX	lbs
14	QØ	lbs/ft^2
15	ALPHA	degrees
16	GAMMA	degrees
17	WEIGHT	lbs
18	LIFT	lbs
19	DRAG	lbs
20	WDOTP	lbs/sec
21	RES1	not used
22	ISPAVG	secs
23	ACC	g's, accel/GA
24	ACCCOM	g's
25	THRUST/DRAG	lbs/lbs

One use of this file is for generating plots of different variables after run completion. The spreadsheet program LOTUS 1-2-3 can be used to generate graphics plots. The program ET0123 on the distribution disk can be used to build a file from this data that can be read by LOTUS.

At the end of the flight, you will be prompted for a printout of the plotter/printer file (presuming you responded with "Y" at start of problem). If you respond with "Y", then you will get an output giving the final run conditions, a copy of the input data files (first-stage and second-stage) and a time-ordered history of all of the above variables. This is the only opportunity you will have to generate this information from within the program. See the sample output at the end of this section. The program was written so that all printed output could fit on standard 8½-inch-wide sprocket-feed computer paper if compressed print (17 characters per inch) is supported by the printer. The code to turn on and turn off the compressed print is compatible with the Epson FX series dot matrix printers. If your printer cannot generate the 17-character/inch compressed print, you will have to use a standard wide carriage printer and wide paper. If your printer will not respond to the compressed print commands generated by the program but can be set by a printer switch to compressed print, you will be able to use 8½-inch paper.

If you are working with a two-stage vehicle, you will be prompted at completion of its first-stage flight that the second-stage input file has been read and the value that was read from the input file. If the file was read correctly and the stopping conditions of the first-stage were satisfactory, then the second-stage run can proceed. If you want to stop, press CTRL-C and the program will exit to DOS. Presuming you wish to proceed, you will need to respond to the prompt for a listing of the second-stage tabular data (nine input data tables). If the response is "Y" the tables will be listed and the program will then run to completion. An "N" response bypasses this printout and starts the second-stage problem. At the end of a second-stage run, you will be prompted for a printout of the plotter/printer file if you had requested one at the start of the problem.

Any time during the run, pressing CTRL-S or the BREAK key will suspend program execution. The program can be resumed from this point by pressing the spacebar. If you wish to terminate the program after suspending it, press the spacebar first to resume it, then press CTRL-C to terminate. The program may be terminated at any

time by pressing CTRL-C. If you had requested a plotter/printer file at the beginning of the run, you will be prompted for a printout of the run up to the point of termination.

3.3 Running ETO within the QUICKBASIC 4.0 Environment

The program was developed and tested using this compiler. It places the user in direct contact with the source code, allowing modifications to be made even within a run. Also, a "scratch space" can be utilized within the source code to temporarily overwrite/replace variables that were read in from the input data file. For example, between runs, fuel weight, launch weight, or other variables might be adjusted if program run results indicate the need. Values written into the "scratch space" later can be moved to the input data file. The Quick BASIC environment allows other files, such as an input data file to be loaded, modified, saved and remain resident, while the main program is still in memory.

To run the program in the Quick BASIC editor/compiler environment, you need to follow the compiler instructions for loading Quick BASIC and ETO13.BAS. After selecting the RUN option and pressing ENTER, the prompts and required responses are exactly like those described previously in this section. Screen output and printer output are identical also. Be sure to use a copy of the distribution disk, not the original. Also, the MS-DOS COMMAND.COM for your computer should be on the disk or default drive. We have found that 100 percent IBM compatibility is needed to use the Quick BASIC editor/compiler environment.

3.4 Input Data Files for a Single- or First-Stage Vehicle

All of the information about the vehicle to be simulated is contained in an input data file (two files for a two-stage vehicle) which is/are opened by the program and the information read and stored in memory. The input file(s) consists of nine data tables plus additional vehicle information. An input file is normally less than 2,000 bytes, (about three quarters of a page).

Input File Information:

The first set of information is two lines of descriptive text about the data file, such as name, type of vehicle, etc. The program reads these two lines as string variables, so any ASCII characters may be used. Each line should be limited to 80 characters. Next follows nine data tables (Tables 1 through 9). See Figure 3-1* and the input files at the end of Section 3.6.

Table 1. Airbreather Specific Impulse

This table is the airbreather specific impulse, ISPA. This table must be constructed as follows:

ISPA, 2
n (number of Mach args), m (number of phi args)
M_1, M_2, M_3, ----------- Mn
$\emptyset_1$, $\emptyset_2$, $\emptyset_3$, ----------- $\emptyset$m

Isp_1, Isp_2, Isp_3, -------- Isp_n at $\emptyset_1$
Isp_1, Isp_2, Isp_3, -------- Isp_n at $\emptyset_2$
Isp_1, Isp_2, Isp_3, -------- Isp_n at $\emptyset$m

The above example is for a 2-D table. For a 1-D table, use the following format:

ISPA, 1
n, number of Mach args.
M_1, M_2, M_3, -------- M_n
Isp_1, Isp_2, Isp_3, -------- Isp_n

The program uses the INPUT# statement so it expects all values to be separated by commas. If more than one line is required to list a group of variables, then the data can be continued on the next line. Do not use a comma as the last character in a line to be continued.

*See page 35.

Table 2. Airbreather Maximum Fuel Equivalence Ratio

The next data field is the airbreather maximum allowable fuel equivalence ratio as a function of Mach number:

This is a 1-D table:
PHIMAX, 1
n, number of Mach args.
M_1, M_2, M_3, -------- M_n
$\emptyset max_1$, $\emptyset max_2$, $\emptyset max_3$, -------- $\emptyset max_n$

Following the same format as the above, the remainder of the tabular data follows:

Table 3. Zero Lift Drag Coefficient

CDØ, 1
n (number of Mach args)
M_1, M_2, M_3, -------- Mn

Table 4. Friction Drag Increment as a Function of Altitude

DELCD, 1
n (number of altitude args)
ho_1, ho_2, ho_3, -------- ho_n
ΔC_{D1}, ΔC_{D2}, ΔC_{D3}, -------- ΔC_{Dn}

Table 5. Lift Curve Slope as a Function of Mach Number

CLALPHA, 1
n (number of Mach args)
M_1, M_2, M_3, -------- Mn
$C_{L\alpha 1}$, $C_{L\alpha 2}$, $C_{L\alpha 3}$, -------- $C_{L\alpha n}$

Table 6. Value of Drag Due to Lift Factor K, as a Function of Mach Number

K, 1

n (number of Mach args)

$M_1, M_2, M_3,$ -------- Mn

$K_1, K_2, K_3,$ -------- Kn

Table 7. Airbreather Inlet Capture Area Ratio vs Mach Number and Angle of Attack

This is most often a 2-D table. The example data file TEST1.DAT, however, shows how it is made effectively a 1-D table by making the Ao/Ac values invariant with alpha.

AØAC, 2

n (number of Mach args), m (number of α args)

$M_1, M_2, M_3,$ -------- Mn

$\alpha_1, \alpha_2, \alpha_3 \ldots\ldots \alpha_m$

$AØAC_1$, $AØAC_2$, $AØAC_3$, -------- AØAC at α_1

$AØAC_1$, $AØAC_2$, $AØAC_3$, -------- AØAC at α_2

.

.

$AØAC_1$, $AØAC_2$, $AØAC_3$, -------- AØAC at α_m

Table 8. Rocket Specific Impulse vs Altitude

ISPR, 1

n (number of altitudes)

$ho_1, ho_2, ho_3,$ -------- ho_n

$Isp_{R1}, Isp_{R2}, Isp_{R3},$ -------- Isp_{Rn}

This table allows the rocket performance to increase with altitude in accordance with the increasing nozzle pressure ratio.

Table 9. Maximum Rocket Propellant Flow Rate vs Mach Number

WDOTPMAX, 1

n (number of Mach args)

$M_1, M_2, M_3,$ -------- Mn

$WDOTPMAX_1$, $WDOTPMAX_2$, $WDOTPMAX_3$, -------- $WDOTPMAX_n$

Note that the data tables do not require uniform increments for the arguments. Note also that a step change in the value of a variable can be used, e.g., a rocket can be turned on at Mach 8 with a propellant flow of 600 lbs/sec by having WDOTPMAX go from 0 to 600 between Mach 8 and 8.01. Linear interpolation is used between data points.

Following the tabular input values the following data items must be entered:

<u>AC, AREF</u>

AC is the airbreather inlet capture area in sq feet

AREF is the vehicle reference area in sq feet.

<u>WLAUNCH, WFUEL, WFINAL, VFINAL, STAGE</u>

WLAUNCH is launch weight, lbs

WFUEL is fuel plus rocket propellant, lbs

WFINAL is burnout weight of stage, lbs

VFINAL is the desired final velocity for this stage. For a single stage to orbit vehicle use a value beyond the polar orbit velocity, say 27,000 fps unless a smaller velocity is desired for a lower inclination orbit.

STAGE is 1 for a single-stage vehicle or the first-stage of a two-stage vehicle.

<u>MØ, HØ, GAMMA, ALPHA, DT, DELPRINT, TIMEX</u>

MØ is the starting Mach number, a value of Ø.ØØ1 is suggested instead of zero for a stationary launch. For an air launch, use the launch Mach number. For a low angle ground launch, use the Mach number at the end of the rail (assuming some cold launch system is used for the initial acceleration). HØ is the launch altitude in feet. Use Ø.ØØ1 instead of a zero value.

GAMMA is the initial flight path angle, degrees. A value of Ø.Ø may be used for a horizontal launch.

ALPHA is the initial angle of attack, degrees. A value of Ø.Ø may be used for a horizontal launch.

DT is the integration interval, secs. The program is designed to use large values, i.e., 2 to 5 secs. If the flight seems to exhibit large, erratic angle of attack excursions in Phase III, then try smaller values of DT.

DELPRINT is the frequency at which integration steps are displayed to the screen. A value of 20 to 40 insures that program execution is not being controlled by the rate that the screen can be updated.

TIMEX is the starting value for the program time in seconds. Ø.Ø may be used to start the problem.

QCOMD, QØMAX, QØMIN, ALPHAMAX, ALPHAMIN

QCOMD is the commanded dynamic pressure during Phase III flight, psf.

QØMAX and QØMIN are values of q greater and lesser, respectively, than QCOMD that the vehicle is limited to.

ALPHAMAX and ALPHAMIN are maximum and minimum values that are permitted during Phases III and IV.

VTAKEOFF, ALPHA2MAX

VTAKEOFF is the takeoff rotation speed in feet/sec.

ALPHA2MAX is the maximum angle of attack permitted during Phase II, the interval from rotation to Phase III flight. Normally this angle has its maximum just as Phase II begins. It becomes the vehicle rotation angle.

LOADFAC, ACCCOMD, GAMMAMAX

LOADFAC is the maximum normal load factor permitted during the Phase II climb, in g's. The angle of attack is permitted to become as large as ALPHA2MAX in order to generate the maximum LOADFAC.

ACCCOMD is the commanded maximum axial acceleration, in g's, during Phases I, II, and III.

GAMMAMAX is maximum flight path angle, in degrees, permitted during any flight phase except Phases IV and V.

FASTOIC

FASTOIC is the stoichiometric fuel-air ratio for the airbreathing propulsion.

VØTEMP, QØFINAL

VØTEMP is the flight velocity, in ft/sec during Phase III at which the vehicle begins to command to a final dynamic pressure QØFINAL. This variation is made linearly between VØTEMP and VFINAL. A combination of VØTEMP and QØFINAL is usually selected so the vehicle will ascend along an approximation of a constant vehicle skin temperature path. Enter a value of 0.0 for VØTEMP and the same value as QCOMD for QØFINAL if constant q climb to VFINAL is desired.

VØCRUISE, SWITCH4, GAMMA4MAX

VØCRUISE is the velocity, in ft/sec, that may be selected for a constant velocity cruise to fuel exhaustion. This is Phase IV. Enter a zero if there is no cruise phase.

SWITCH4 has a value of either 0 or 1. Zero selects cruise at constant altitude. The vehicle maintains the altitude it attained during the Phase III climb. A value of 1 selects cruise at maximum lift-to-drag ratio. Enter a zero if there is no cruise.

GAMMA4MAX is the maximum flight path angle (in degrees) permitted during a maximum L/D cruise (SWITCH4=1). A small value, in the range of 0.20 to 0.50 degrees is suggested. The value should be chosen so that the vehicle performs a fairly smooth climb/cruise only occasionally encountering the limiting value after experiencing some flight path oscillations early in the cruise. This variable has no

significance for a constant altitude cruise (SWITCH4=0) and a value of zero should be entered.

FLAG5, VØ5, ALPHA5MAX, GAMMA5, ACCCOMD5

These five variables are used to prescribe a pullup and ballistic ascent after completing part of the ascent after another phase. This is Phase V.

FLAG5 can have a value of 0 or 1. Zero means that a ballistic ascent is not going to be used. In this case, enter zeros for all the remaining variables on this line. If FLAG5 is set to 1, then enter values for the following:

VØ5 is the velocity, ft/sec, at which pullup is to begin. ALPHA5MAX is the commanded angle of attack, in degrees, to be employed during pullup.

GAMMA5 is the flight path angle, in degrees, to be achieved during the pullup. As soon as this flight path angle is achieved, angle of attack is commanded to zero to permit the flight to continue ballistically.

ACCCOMD5 is the axial acceleration, g's, commanded during Phase V. Thrust is reduced if required to avoid exceeding this commanded acceleration.

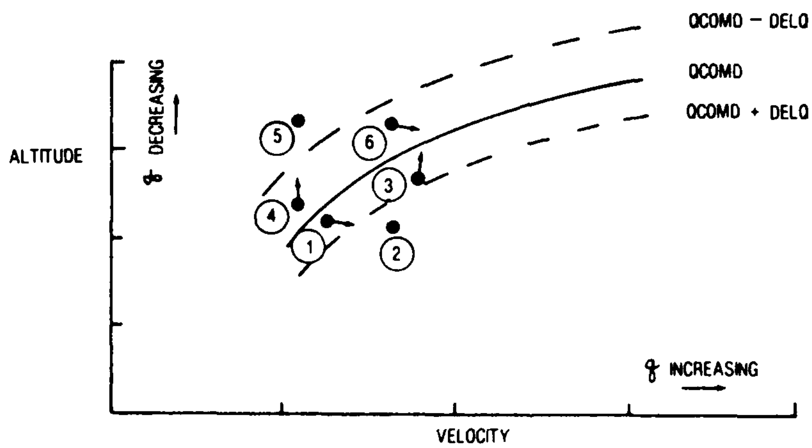
A1, A2, B1, C1, C2, D1, DELO, VØ2, HØ2

The first seven variables are adjustment coefficients used in the Phase III navigation algorithm, GUID13 (see Figure 2-4 for flow chart of control logic). These coefficients are used to set the sensitivity of angle of attack (α) while the vehicle is trying to follow a constant dynamic pressure (q) path. Suggested values are, in order:

Ø.1, Ø.1, Ø.ØØ9, Ø.1, 2.Ø, Ø.1, 5Ø.Ø

As a further suggestion, set DELQ at about 5 percent of the commanded Phase III dynamic pressure, QCOMD. Some work has been done to implement a simpler Phase III algorithm, GUID14, (included on distribution disk as GUID14.BAS), which has constants "hardwired" into the code. Whether you are using GUID13 or GUID14, this is an area where some experimentation may be needed to obtain smooth constant q trajectory paths.

If the vehicle deviates from a smooth constant q trajectory, the following information will assist you in making the necessary changes. There are six possible deviations that may occur as shown in the following sketch:



Note that the value DELQ is used to determine if the vehicle reaches deviations 2 and 5. In order to correct each of these deviations (1 to 6), α must be changed such that the vehicle returns to the desired q path. For each different deviation, there is a different α sensitivity that is required to return to the desired dynamic pressure path. If the vehicle is having difficulty recovering from any of the deviations, identify the deviation condition that exists and slightly vary the corresponding adjustment coefficient.

The adjustment coefficients affect the angle of attack in the following manner.

DESCRIPTION	"q is"	REACTION	CALCULATION
(1)	Higher than QCOMD & Increasing	Increase α	$\alpha = \alpha + A_1 + A_2 \cdot \frac{dq}{dt}$
(2)	Higher than QCOMD & Outside DELQ	Increase α	

DESCRIPTION "q is"		REACTION	CALCULATION
(3)	Higher than QCOMD & Decreasing	Decrease α	$\alpha = \alpha - B1 \cdot \frac{Q-QCOMD}{QCOMD}$
(4)	Lower than QCOMD & Decreasing	Decrease α	$\alpha = \alpha - C1 - C2 \cdot \frac{dq}{dt}$
(5)	Lower than QCOMD & Outside DELQ	Decrease α	
(6)	Lower than QCOMD & Increasing	Increase α	$\alpha = \alpha + D1 \cdot \frac{Q-QCOMD}{QCOMD}$

Note that in conditions 3 and 6 the control logic is attempting to prevent overshooting QCOMD.

VØ2 and HØ2 are a velocity (ft/sec) and altitude (feet) pair which defines the end of the Phase II constant load factor climb. The vehicle is not permitted to exceed VØ2 until the altitude, HØ2 has been exceeded. For an aircraft, VØ2 values of 700 to 900 ft/sec are usually chosen. HØ2 can range from 10,000 feet to 30,000 feet. A combination of HØ2, VØ2 values is usually chosen that would be consistent with an optimum subsonic-transonic climb, such as would be prescribed by a Rutowski path optimization method.

3.5 Input Data Files for a Second-Stage Vehicle

The nine data tables described in the previous section are required and are used in the same manner for a second-stage vehicle.

The remaining values required are

AC, AREF

WEIGHT, WFUEL, WFINAL, VFINAL, STAGE

DT, DELPRINT

QØCOMD, QØMAX, QØMIN, ALPHAMAX, ALPHAMIN

LOADFAC, ACCCOMD, GAMMAMAX

FASTIOC

VØTEMP, QØFINAL

VØCRUISE, SWITCH4, GAMMA4MAX

FLAG5, VØ5, ALPHA5MAX, GAMMA5, ACCCOMD5

A1, A2, B1, C1, C2, D1, DELQ, VØ2, HØ2

Note that the variables VTAKEOFF and ALPHA2MAX are not used, and since the flight is continuing from the previous stage, starting values of MØ, HØ, GAMMA, ALPHA, and TIMEX are not input. When the second-stage data file is loaded, its values overwrite the first-stage data and current values of appropriate program variables. As an example, the current value of WEIGHT is replaced by the value of WEIGHT read from the second-stage file.

3.6 Sample Input Data Files and Program Output

This section describes the example input data files contained on the distribution disk. Each example illustrates features typically used in combination. The example input data files are completely arbitrary and are not intended to represent any known or projected vehicle or level of technology. No inference should be made that the example vehicles typify current or projected capabilities.

The file TEST1.DAT is the simplest example. It is a highly speculative single-stage-to-orbit horizontal takeoff vehicle having all-airbreathing propulsion to orbital velocity. The vehicle is commanded to fly to the orbital condition at a constant 1,000 lb/ft^2 dynamic pressure. The vehicle has a gross takeoff weight of 100,000 lbs and has 60,000 lbs of fuel.

The file STG1.DAT and STG2.DAT define a two-stage vehicle intended to fly to orbit from a horizontal takeoff. The first stage uses airbreathing propulsion up to the Mach 8 staging point. The second stage operates with rocket propulsion from staging to orbit. The gross takeoff weight of the vehicle is 400,000 lbs, of which 150,000 lbs is first-stage fuel. The second stage weighs 150,000 lbs at staging, of which 120,000 lbs is propellant. The empty weight of the first stage is 100,000 lbs. The first stage accelerates along a constant 1,000 lb/ft^2 dynamic pressure trajectory to the staging condition.

The file STG2BAL.DAT may be used as an alternative to STG2.DAT, above, as a second stage. This vehicle is identical to STG2.DAT but employs a pullup after staging and ballistic ascent to orbit.

The file CRUZ2.DAT is a Mach 4 cruise aircraft. It has a gross takeoff weight of 400,000 lbs, of which 170,000 lbs is fuel. It employs airbreathing propulsion in all flight phases. It cruises at maximum L/D.

A maximum L/D cruise may experience oscillations in altitude and flight path angle if the vehicle does not start the cruise phase at lift nearly equal to weight and at a very low flight path angle. The code includes an algorithm that smooths the climb/cruise flight path by reducing angle of attack whenever GAMMA4MAX is exceeded and whenever lift becomes less than weight. Also, if the flight path angle becomes negative, angle of attack is increased. This algorithm seems to produce fairly smooth climb/cruise paths once it has forced the vehicle to a fairly steady lift equal weight condition.

The program tests if the vehicle input data files have been read correctly. In starting the program, the following message will appear on the screen just above the prompt for creation of a plotter/printer file:

1ST/SINGLE (OR 2ND) STAGE INPUT FILE HAS BEEN READ. LAST ITEM IN INPUT FILE IS HØ2, VALUE READ FROM FILE IS 18ØØØ. IF THIS VALUE IS INCORRECT, INPUT FILE IS IN ERROR.

In all of the example vehicle files, the value of HØ2 is 18,ØØØ. The screen message above indicates that the program has read all of the data correctly. If you see an unexpected value, this means that the number of data items in the file exceeds the number that the program was directed to read. Check to see that the number of entries in the nine data tables correspond correctly with the declared number and that the data entered after the tabular input is correct. The problem will be found to be too many entries. A duplicate line of data or extra commas is the usual cause. An error of too many data items will not prevent the program from running, but erroneous results will immediately be apparent. The other type of error, not enough data items, will not permit the program to get as far as the prompt above. An error message "Input past end" and program termination will occur if this error is present. In this case the program is looking for data which is not present. A line of data items omitted, or a lacking comma are the usual causes. The program is quite exacting in its requirement that the input file correspond precisely to the declared number of items to be read. Values, separated by commas, must appear exactly in the prescribed format. If an option is not to be employed, or a zero value is planned for a variable, a numerical entry must be present (even a zero).

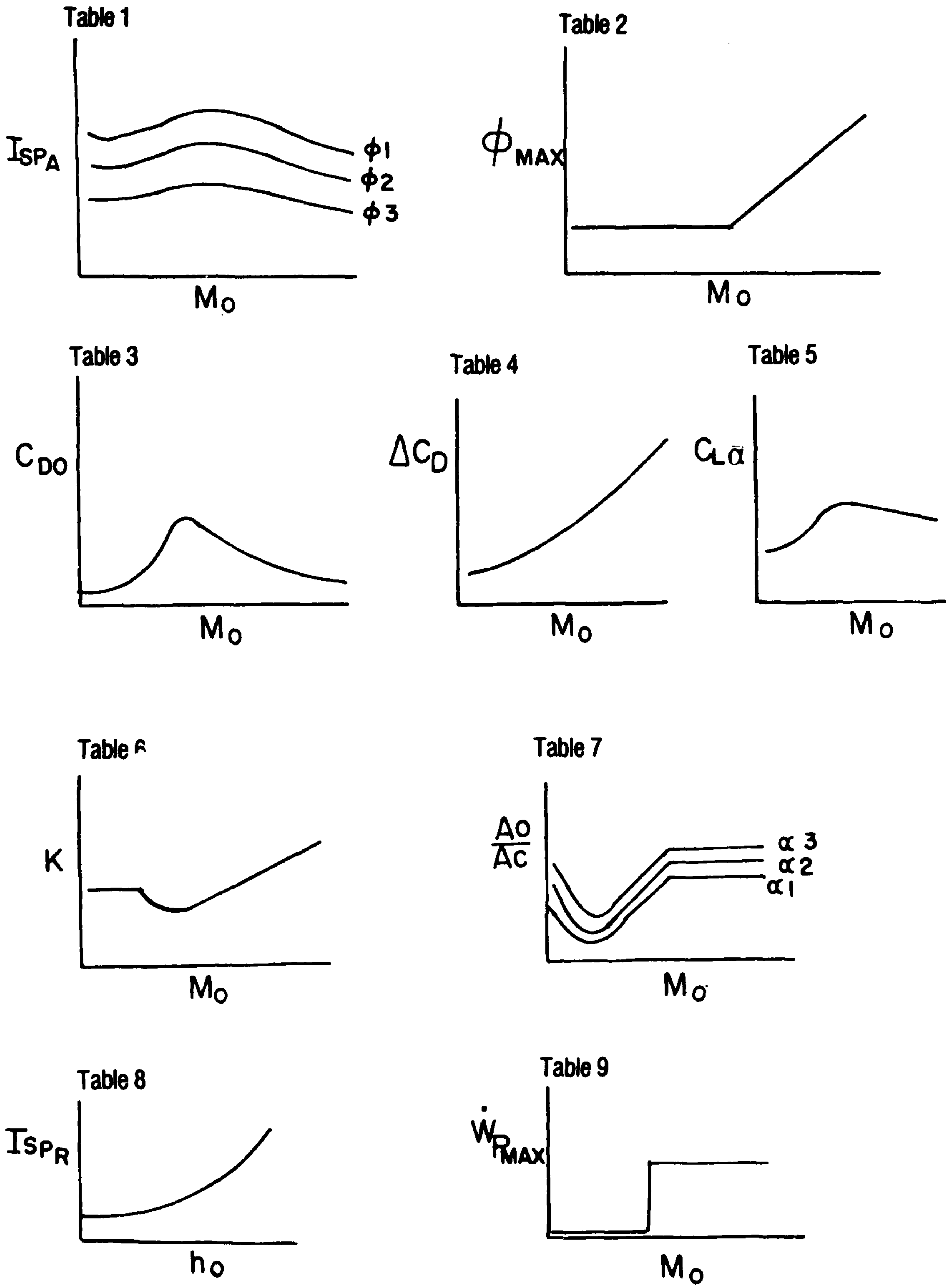


Figure 3-1. Tabular Input Data

Sample Input Data Files

```
INPUT DATA FILE "TEST1.DAT" 1 SEPT88                          PROGRAM "ETO"
A/B TO ORBIT AT CONSTANT Q=1ØØØ
ISPA,2
12,3
Ø.Ø ,     .5 ,.9 ,1.5 ,2.Ø ,2.5 ,3.Ø ,4.Ø ,8.Ø ,12.Ø ,2Ø.Ø ,28.Ø
Ø.1 ,1.Ø ,1Ø.Ø
4ØØØ,      4ØØØ,35ØØ,33ØØ,365Ø,375Ø,37ØØ,35ØØ,3ØØØ,18ØØ,8ØØ,6ØØ
3ØØØ,      3ØØØ,3ØØØ,33ØØ,365Ø,375Ø,37ØØ,35ØØ,3ØØØ,18ØØ,8ØØ,6ØØ
5ØØ,       5ØØ, 5ØØ, 5ØØ, 5ØØ, 5ØØ, 5ØØ, 5ØØ, 6ØØ,5ØØ,4ØØ,4ØØ
PHIMAX,1
1Ø
Ø.Ø ,Ø.8 ,1.Ø ,1.4 ,2.Ø,8.Ø ,16.Ø ,18.Ø ,25.Ø ,28.Ø
1.Ø ,1.Ø ,1.Ø ,1.Ø ,1.Ø,1.Ø , 3.Ø , 3.Ø , 5.Ø , 5.Ø
CDØ,1
15
Ø.Ø, .4 , .8 , .9 ,1.Ø ,1.1 ,1.2 ,1.5 ,2.Ø ,3.Ø ,5.Ø ,1Ø.Ø ,15.Ø ,2Ø.Ø ,25.Ø
Ø.Ø,.Ø1Ø,.Ø1Ø,.Ø1Ø,.Ø23,.Ø22,.Ø18,.Ø16,.Ø13,.ØØ8,.ØØ4,.ØØ2,.ØØ19,.ØØ17,.ØØ15
DELCD,1
5
Ø.Ø,5ØØØØ.,1ØØØØØ.,15ØØØØ.,3ØØØØØ.
Ø.Ø,.ØØØ6 ,.ØØ2    ,.ØØ1    ,.ØØ3Ø
CLALPHA,1
1Ø
Ø.Ø,         1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.Ø4Ø,      .Ø4Ø,.Ø39 ,.Ø24, .Ø2Ø, .Ø18,.Ø17,.Ø16, .Ø1Ø ,.Ø1Ø
K,1
1Ø
Ø.Ø,         1.5  ,2.Ø , 3.Ø  ,4.Ø ,5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.4ØØ   ,   Ø.4ØØ ,.44Ø,.73Ø ,.87Ø,.97Ø,1.Ø25,1.Ø9Ø ,1.745 ,1.745
AØAC,2
1Ø,3
Ø.Ø   , Ø.Ø1,  Ø.1 ,    Ø.5 ,     Ø.8 ,      1.Ø ,     1Ø.Ø , 15,  15.Ø1     ,28.Ø
-1Ø.Ø ,Ø.Ø ,1Ø.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,  .5Ø,     1.Ø     ,1.Ø,1.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,  .5Ø,     1.Ø     ,1.Ø,1.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,  .5Ø,     1.Ø     ,1.Ø,1.Ø
ISPR,1
4
Ø.Ø ,   3ØØØØ.Ø   , 8ØØØØ.Ø   ,35ØØØØ.Ø
39Ø    ,44Ø       ,445        ,46Ø
WDOTPMAX,1
4
Ø.Ø   , 8.Ø   ,8.Ø1    ,28.Ø
Ø.Ø   , Ø.Ø   ,Ø.Ø     ,Ø.Ø
1ØØ.Ø   , 1875.Ø
1ØØØØØ.,6ØØØØ.,4ØØØØ. ,27ØØØ ,1
Ø.ØØ1 ,Ø.ØØ1 ,Ø.Ø ,Ø.Ø ,    5.Ø ,   1Ø.ØØ , Ø.Ø
 1ØØØ.Ø,11ØØ.Ø,9ØØ.Ø, 5.Ø , -5.Ø
375.Ø ,5.Ø
2.ØØ,  Ø.5Ø , 8.Ø
Ø.Ø292
Ø  ,1ØØØ
Ø ,Ø ,Ø
Ø ,Ø ,Ø ,Ø ,Ø
 .1 , .1 , .ØØ9, .1, 2.Ø  , .1  ,5Ø  ,9ØØ  ,18ØØØ
```

```
INPUT DATA FILE "  STG1.DAT"/26JUNE88/153Ø/CONST Q TO ORBIT  /PROGRAM"ETO2"
ALL AIRBREATHING PROPULSION TO MACH 8 /FIRST STAGE
ISPA,2
13,3
Ø.Ø , Ø.5 ,1.Ø, 1.5 ,2.Ø ,2.5 ,3.Ø ,4.Ø, 5.Ø , 5.1  , 8.Ø , 15.Ø ,  26.Ø
Ø.1 ,1.Ø ,1Ø.Ø
45ØØ, 44ØØ,42ØØ,4ØØØ,37ØØ,35ØØ,32ØØ,28ØØ,24ØØ, 24ØØ,  23ØØ, 1ØØØ,   6ØØ
45ØØ, 44ØØ,42ØØ,4ØØØ,37ØØ,35ØØ,32ØØ,28ØØ,24ØØ, 24ØØ,  23ØØ, 1ØØØ,   5ØØ
1ØØØ, 1ØØØ,1ØØØ,1ØØØ,1ØØØ,1ØØØ,1ØØØ, 8ØØ, 6ØØ, 6ØØ,   6ØØ,  45Ø,   45Ø
PHIMAX,1
12
Ø.Ø ,Ø.8 ,1.Ø ,1.4 ,2.Ø, 5.Ø, 5.1, 8.Ø ,16.Ø ,18.Ø ,25.Ø ,28.Ø
1.Ø ,1.Ø ,1.Ø ,1.Ø ,1.Ø, 1.Ø, 1.Ø , 1.Ø, 3.Ø , 5.Ø , 5.Ø , 5.Ø
CDØ,1
15
Ø.Ø,.4 ,.8 ,.9 ,1.Ø ,1.1 ,1.2 ,1.5 ,2.Ø ,3.Ø ,5.Ø ,1Ø.Ø ,15.Ø ,2Ø.Ø ,25.Ø
Ø.Ø,.ØØ9,.ØØ9,.ØØ9,.Ø23,.Ø22,.Ø18,.Ø16,.Ø13,.ØØ8,.ØØ4,.ØØ2,.ØØ19,.ØØ17,.ØØ15
DELCD,1
5
Ø.Ø,5ØØØØ.,1ØØØØØ.,15ØØØØ.,2ØØØØØ.
Ø.Ø,.ØØØ6 ,.ØØ2   ,.ØØ1   ,.ØØ2Ø
CLALPHA,1
1Ø
Ø.Ø,        1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.Ø43,     .Ø43,.Ø391,.Ø24, .Ø2Ø, .Ø18,.Ø17,.Ø16, .Ø1Ø ,.Ø1Ø
K,1
1Ø
Ø.Ø,        1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.4Ø7     , Ø.4Ø7,.447,.728,.873 ,.97Ø ,1.Ø26,1.Ø91 ,1.745,1.745
AØAC,2
9,3
Ø.Ø  , Ø.Ø1,  Ø.1 ,    Ø.5 ,     Ø.8 ,     1.Ø ,     1Ø.Ø ,     25.Ø      ,28.Ø
-1Ø.Ø ,Ø.Ø ,1Ø.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,   .51,     1.Ø      ,1.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,   .51,     1.Ø      ,1.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,   .51,     1.Ø      ,1.Ø
ISPR,1
4
Ø   ,          3ØØØØ,      8ØØØØ,       35ØØØØ
39Ø ,          44Ø,        445,         46Ø
WDOTPMAX,1
4
Ø.Ø ,         8.Ø ,        8.Ø1 ,        28
Ø.Ø ,         Ø.Ø ,        Ø.Ø  ,        Ø.Ø
4ØØ.Ø   , 55ØØ.Ø
4ØØØØØ ,15ØØØØ,25ØØØØ ,8ØØØ , 1
Ø.ØØ1 ,Ø.ØØ1 ,Ø.Ø ,Ø.Ø ,     2.Ø,    1Ø.Ø , Ø.Ø
15ØØ.Ø,16ØØ.Ø,14ØØ.Ø, 5.Ø , -5.Ø
39Ø.Ø ,15.Ø
2.ØØ,   .5Ø , 8.Ø
Ø.Ø292
Ø       , 1ØØØ
Ø ,Ø ,Ø
Ø    ,Ø    ,Ø     ,Ø    ,Ø
 .1  ,.1  ,  .ØØ9  , .1  ,  2.Ø  ,  .1  , 5Ø.Ø,9ØØ , 18ØØØ
```

```
INPUT DATA FILE "STG2.DAT"/26JUNE88/153Ø/CONSTANT Q TO ORBIT /PROGRAM"ETO2"
ALL ROCKET SECOND STAGE
ISPA,2
13,3
Ø.Ø , Ø.5 ,1.Ø, 1.5 ,2.Ø ,2.5 ,3.Ø ,4.Ø, 5.Ø , 5.1  , 8.Ø , 15.Ø ,  26.Ø
Ø.1 ,1.Ø ,1Ø.Ø
45ØØ, 44ØØ,42ØØ,4ØØØ,37ØØ,35ØØ,32ØØ,28ØØ,24ØØ, 24ØØ,  23ØØ, 1ØØØ,   6ØØ
45ØØ, 44ØØ,42ØØ,4ØØØ,37ØØ,35ØØ,32ØØ,28ØØ,24ØØ, 24ØØ,  23ØØ, 1ØØØ,   5ØØ
1ØØØ, 1ØØØ,1ØØØ,1ØØØ,1ØØØ,1ØØØ,1ØØØ, 8ØØ, 6ØØ, 6ØØ,   6ØØ,  45Ø,    45Ø
PHIMAX,1
12
Ø.Ø ,Ø.8 ,1.Ø ,1.4 ,2.Ø, 5.Ø, 5.1, 8.Ø ,16.Ø ,18.Ø ,25.Ø ,28.Ø
Ø.Ø ,Ø.Ø ,Ø.Ø ,Ø.Ø ,Ø.Ø, Ø.Ø, Ø.Ø , Ø.Ø, Ø.Ø , Ø.Ø , Ø.Ø , Ø.Ø
CDØ,1
15
Ø.Ø,.4 ,.8 ,.9 ,1.Ø ,1.1 ,1.2 ,1.5 ,2.Ø ,3.Ø ,5.Ø ,1Ø.Ø ,15.Ø ,2Ø.Ø ,25.Ø
Ø.Ø,.ØØ9,.ØØ9,.ØØ9,.Ø23,.Ø22,.Ø18,.Ø16,.Ø13,.ØØ8,.ØØ4,.ØØ2,.ØØ19,.ØØ17,.ØØ15
DELCD,1
5
Ø.Ø,5ØØØØ.,1ØØØØØ.,15ØØØØ.,2ØØØØØ.
Ø.Ø,.ØØØ6 ,.ØØ2   ,.ØØ1   ,.ØØ2Ø
CLALPHA,1
1Ø
Ø.Ø,         1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.Ø43,      .Ø43,.Ø391,.Ø24, .Ø2Ø, .Ø18,.Ø17,.Ø16, .Ø1Ø ,.Ø1Ø
K,1
1Ø
Ø.Ø,         1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.4Ø7      , Ø.4Ø7,.447,.728,.873 ,.97Ø ,1.Ø26,1.Ø91 ,1.745,1.745
AØAC,2
9,3
Ø.Ø  , Ø.Ø1, Ø.1 ,   Ø.5 ,    Ø.8 ,    1.Ø ,    1Ø.Ø ,    25.Ø      ,28.Ø
-1Ø.Ø ,Ø.Ø ,1Ø.Ø
8Ø.Ø,  2.ØØ, Ø.5Ø,  Ø.15,    .15,     .15     ,  .51,   1.Ø     ,1.Ø
8Ø.Ø,  2.ØØ, Ø.5Ø,  Ø.15,    .15,     .15     ,  .51,   1.Ø     ,1.Ø
8Ø.Ø,  2.ØØ, Ø.5Ø,  Ø.15,    .15,     .15     ,  .51,   1.Ø     ,1.Ø
ISPR,1
4
Ø   ,          3ØØØØ,      8ØØØØ,       35ØØØØ
39Ø ,          44Ø,        445,         46Ø
WDOTPMAX,1
4
Ø.Ø ,         8.Ø ,        8.Ø1 ,        28
6ØØ ,         6ØØ ,        6ØØ  ,        6ØØ
1ØØ.Ø  , 15ØØ.Ø
15ØØØØ ,12ØØØØ, 3ØØØØ.Ø ,27ØØØ  , 2
 2.Ø,   1Ø.Ø
1ØØØ.Ø,11ØØ.Ø,9ØØ.Ø, 5.Ø , -5.Ø
2.ØØ,  1.ØØ , 8.Ø
Ø.Ø292
Ø      , 1ØØØ
Ø ,Ø ,Ø
Ø    ,Ø    ,Ø    ,Ø    ,Ø
 .1   ,.1  ,  .ØØ9  , .1  ,  2.Ø  ,  .1  , 5Ø.Ø,9ØØ , 18ØØØ
```

```
INPUT DATA FILE "STG2BAL.DAT"/29DEC88/BALLISTIC ASCENT TO ORBIT/"ETO"
ALL ROCKET SECOND STAGE
ISPA,2
13,3
Ø.Ø , Ø.5 ,1.Ø, 1.5 ,2.Ø ,2.5 ,3.Ø ,4.Ø, 5.Ø , 5.1  , 8.Ø , 15.Ø ,  26.Ø
Ø.1 ,1.Ø ,1Ø.Ø
45ØØ, 44ØØ,42ØØ,4ØØØ,37ØØ,35ØØ,32ØØ,28ØØ,24ØØ, 24ØØ,  23ØØ, 1ØØØ,   6ØØ
45ØØ, 44ØØ,42ØØ,4ØØØ,37ØØ,35ØØ,32ØØ,28ØØ,24ØØ, 24ØØ,  23ØØ, 1ØØØ,   5ØØ
1ØØØ, 1ØØØ,1ØØØ,1ØØØ,1ØØØ,1ØØØ,1ØØØ, 8ØØ, 6ØØ, 6ØØ,   6ØØ, 45Ø,   45Ø
PHIMAX,1
12
Ø.Ø ,Ø.8 ,1.Ø ,1.4 ,2.Ø, 5.Ø, 5.1, 8.Ø ,16.Ø ,18.Ø ,25.Ø ,28.Ø
Ø.Ø ,Ø.Ø ,Ø.Ø ,Ø.Ø ,Ø.Ø, Ø.Ø, Ø.Ø , Ø.Ø, Ø.Ø , Ø.Ø , Ø.Ø , Ø.Ø
CDØ,1
15
Ø.Ø,.4 ,.8 ,.9 ,1.Ø ,1.1 ,1.2 ,1.5 ,2.Ø ,3.Ø ,5.Ø ,1Ø.Ø ,15.Ø ,2Ø.Ø ,25.Ø
Ø.Ø,.ØØ9,.ØØ9,.ØØ9,.Ø23,.Ø22,.Ø18,.Ø16,.Ø13,.ØØ8,.ØØ4,.ØØ2,.ØØ19,.ØØ17,.ØØ15
DELCD,1
5
Ø.Ø,5ØØØØ.,1ØØØØØ.,15ØØØØ.,2ØØØØØ.
Ø.Ø,.ØØØ6 ,.ØØ2   ,.ØØ1   ,.ØØ2Ø
CLALPHA,1
1Ø
Ø.Ø,        1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.Ø43,     .Ø43,.Ø391,.Ø24, .Ø2Ø, .Ø18,.Ø17,.Ø16, .Ø1Ø ,.Ø1Ø
K,1
1Ø
Ø.Ø,        1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.4Ø7    , Ø.4Ø7,.447,.728,.873 ,.97Ø ,1.Ø26,1.Ø91 ,1.745,1.745
AØAC,2
9,3
Ø.Ø  , Ø.Ø1, Ø.1 ,   Ø.5 ,    Ø.8 ,    1.Ø ,    1Ø.Ø ,    25.Ø     ,28.Ø
-1Ø.Ø ,Ø.Ø ,1Ø.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,   .51,    1.Ø     ,1.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,   .51,    1.Ø     ,1.Ø
8Ø.Ø,   2.ØØ,  Ø.5Ø,   Ø.15,     .15,      .15     ,   .51,    1.Ø     ,1.Ø
ISPR,1
4
Ø   ,           3ØØØØ,      8ØØØØ,       35ØØØØ
39Ø ,           44Ø,        445,         46Ø
WDOTPMAX,1
4
Ø.Ø ,          8.Ø ,        8.Ø1 ,        28
6ØØ ,          6ØØ ,        6ØØ  ,        6ØØ
1ØØ.Ø  , 15ØØ.Ø
15ØØØØ ,12ØØØØ, 3ØØØØ.Ø ,27ØØØ  , 2
  1.Ø,    5.Ø
1ØØØ.Ø,11ØØ.Ø,9ØØ.Ø, 5.Ø , -5.Ø
2.ØØ,  1.ØØ , 8.Ø
Ø.Ø292
Ø      , 1ØØØ
Ø ,Ø ,Ø
1   ,8ØØØ ,1Ø   ,1.5 ,2
 .1  ,.1  ,  .ØØ9  , .1  ,  2.Ø  ,  .1  , 5Ø.Ø,9ØØ , 18ØØØ
```

```
INPUT DATA FILE "  CRUZ2.DAT"/24AUG87/ØØ45/M 4 CRUISE          PROGRAM "ETO"
AB                          SINGLE STAGE
ISPA,2
14,3
Ø.Ø , Ø.5 ,1.Ø, 1.5 ,2.Ø ,2.5 ,3.Ø ,4.Ø, 5.Ø , 5.1  , 8.Ø , 15.Ø,15.Ø1,28.Ø
Ø.1 ,1.Ø ,1Ø.Ø
45ØØ, 44ØØ,42ØØ,4ØØØ,37ØØ,35ØØ,32ØØ,28ØØ,24ØØ, 24ØØ,  23ØØ, 1ØØØ, Ø.Ø ,Ø.Ø
45ØØ, 44ØØ,42ØØ,4ØØØ,37ØØ,35ØØ,32ØØ,28ØØ,24ØØ, 24ØØ,  23ØØ, 1ØØØ, Ø.Ø ,Ø.Ø
1ØØØ, 1ØØØ,1ØØØ,1ØØØ,1ØØØ,1ØØØ,1ØØØ, 8ØØ, 6ØØ, 6ØØ,   6ØØ,  45Ø , Ø.Ø ,Ø.Ø
PHIMAX,1
11
Ø.Ø ,Ø.8 ,1.Ø ,1.4 ,2.Ø, 5.Ø, 5.1, 8.Ø ,15.Ø ,15.Ø1,28.Ø
1.Ø ,1.Ø ,1.Ø ,1.Ø ,1.Ø, 1.Ø, 1.Ø , 1.Ø, 3.Ø , Ø.Ø , Ø.Ø
CDØ,1
15
Ø.Ø,.4 ,.8 ,.9 ,1.Ø ,1.1 ,1.2 ,1.5 ,2.Ø ,3.Ø ,5.Ø ,1Ø.Ø ,15.Ø ,2Ø.Ø ,25.Ø
Ø.Ø,.ØØ9,.ØØ9,.ØØ9,.Ø23,.Ø22,.Ø18,.Ø16,.Ø13,.ØØ8,.ØØ4,.ØØ2,.ØØ19,.ØØ17,.ØØ15
DELCD,1
5
Ø.Ø,5ØØØØ.,1ØØØØØ.,15ØØØØ.,2ØØØØØ.
Ø.Ø,.ØØØ6 ,.ØØ2   ,.ØØ1   ,.ØØ2Ø
CLALPHA,1
1Ø
Ø.Ø,        1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.Ø43,     .Ø43,.Ø391,.Ø24, .Ø2Ø, .Ø18,.Ø17,.Ø16, .Ø1Ø ,.Ø1Ø
K,1
1Ø
Ø.Ø,        1.5 ,2.Ø ,3.Ø , 4.Ø , 5.Ø  ,6.Ø  ,8.Ø  ,15.Ø  ,25.Ø
Ø.4Ø6  ,   Ø.4Ø6,.446,.727,.873, .97 , 1.Ø27,1.Ø91,1.745,1.745
AØAC,2
1Ø,3
Ø.Ø  , Ø.Ø1, Ø.1 ,   Ø.5 ,    Ø.8 ,    1.Ø ,    1Ø.Ø,  15.Ø,15.Ø1,28.Ø
-1Ø.Ø ,Ø.Ø ,1Ø.Ø
8Ø.Ø,  2.ØØ, Ø.5Ø,  Ø.15,    .15,     .15     ,  .51,1.Ø ,Ø.Ø,  Ø.Ø
8Ø.Ø,  2.ØØ, Ø.5Ø,  Ø.15,    .15,     .15     ,  .51,1.Ø ,Ø.Ø,  Ø.Ø
8Ø.Ø,  2.ØØ, Ø.5Ø,  Ø.15,    .15,     .15     ,  .51,1.Ø ,Ø.Ø,  Ø.Ø
ISPR,1
4
Ø,     3ØØØØ,     8ØØØØ,     35ØØØØ
39Ø,   44Ø,       445,       46Ø
WDOTPMAX,1
4
Ø.Ø,    8.ØØ,      8.Ø1,      28.Ø
  Ø.Ø,    Ø.Ø,    6ØØ.Ø,      6ØØ.Ø
4ØØ.Ø  , 55ØØ.Ø
4ØØØØØ ,17ØØØØ,23ØØØØ  , 27ØØØ  , 1
Ø.ØØ1 ,Ø.ØØ1 ,Ø.Ø ,Ø.Ø ,    2.Ø,    2Ø.Ø , Ø.Ø
 8ØØ.Ø, 85Ø.Ø, 75Ø.Ø, 5.Ø , -5.Ø
39Ø.Ø ,15.Ø
2.ØØ,   .5Ø , 8.Ø
Ø.Ø292
Ø      , 1ØØØ
 4ØØØ , 1  , Ø.5
Ø     ,  Ø     ,   Ø    ,    Ø    ,    Ø
 .1  ,.1  ,  .ØØ9  , .1  ,  2.Ø  ,  .1  , 5Ø.Ø,9ØØ , 18ØØØ
```

3.7 Sample Program Output

The following sample output is for the two-stage vehicle defined by the data files STG1.DAT and STG2.DAT. The integration interval was 2 seconds. A FREQ value of 50 was used; thus the output is for every 100 seconds of flight.

01-29-1989 20:17:42

```
1ST/SINGLE STAGE INPUT DATA FILE: STG1.DAT
2ND STAGE INPUT DATA FILE       : STG2.DAT

PLOTTER/PRINTER FILE NAME       : G:2STG.OUT

-------PROGRAM ETO2 RUN SUMMARY---------
    LAUNCH WEIGHT =   400000.00 LBS.
     FINAL WEIGHT  =    38045.15 LBS.
FUEL/PROPELLANT REMAINING =     8045.15 LBS.

        VEHICLE REACHED ORBITAL VELOCITY.

FLIGHT TIME     =     3094.71 SECS.
FINAL VELOCITY =    25858.04 FT/SEC
FINAL ALTITUDE =  152301.95 FT
FINAL Q0        =     1049.85 PSF

AVERAGE ISP     =     1024.11 SEC
MEAN ISP        =      359.19 SEC
```

FINAL CONDITIONS ARE SHOWN BELOW:

TIME	V0	H0	R	ISP	ISPEFF	ISPMEAN	WDOTF	WDOTA	PHI	THRUST	THRUSTCOM	THRUSTMAX
	Q0	ALPHA	GAMMA	WEIGHT	LIFT	DRAG	WDOTP	RES1	ISPAVG	ACCEL-G'S	ACCCOM	T/D
3094.71	25858.04	152301.95	4447.27	449.02	282.09	359.19	0.00	0.00	0.00	29886.49	29886.49	269409.94
3094.71	1049.85	-5.00	0.08	38045.15	-78738.77	10941.28	66.56	0.00	1024.11	0.50	0.50	2.73

```
INPUT DATA FILE "  STG1.DAT"/26JUNE88/1530/CONST Q TO ORBIT  /PROGRAM"ETO2"
ALL AIRBREATHING PROPULSION TO MACH 8 /FIRST STAGE
ISPA,2
13,3
0.0 , 0.5 ,1.0, 1.5 ,2.0 ,2.5 ,3.0 ,4.0, 5.0 , 5.1  , 8.0 , 15.0 ,  26.0
0.1 ,1.0 ,10.0
4500, 4400,4200,4000,3700,3500,3200,2800,2400, 2400,  2300, 1000,   600
4500, 4400,4200,4000,3700,3500,3200,2800,2400, 2400,  2300, 1000,   500
1000, 1000,1000,1000,1000,1000,1000, 800, 600, 600,   600,  450,    450
PHIMAX,1
12
0.0 ,0.8 ,1.0 ,1.4 ,2.0, 5.0, 5.1, 8.0 ,16.0 ,18.0 ,25.0 ,28.0
1.0 ,1.0 ,1.0 ,1.0 ,1.0, 1.0, 1.0 , 1.0, 3.0 , 5.0 , 5.0 , 5.0
CD0,1
15
0.0,.4 ,.8 ,.9 ,1.0 ,1.1 ,1.2 ,1.5 ,2.0 ,3.0 ,5.0 ,10.0 ,15.0 ,20.0 ,25.0
0.0,.009,.009,.009,.023,.022,.018,.016,.013,.008,.004,.002,.0019,.0017,.0015
DELCD,1
5
0.0,50000.,100000.,150000.,200000.
0.0,.0006 ,.002   ,.001   ,.0020
CLALFHA,1
10
0.0,         1.5 ,2.0 ,3.0 , 4.0 , 5.0  ,6.0  ,8.0  ,15.0  ,25.0
0.043,      .043,.0391,.024, .020, .018,.017,.016, .010 ,.010
K,1
10
0.0,         1.5 ,2.0 ,3.0 , 4.0 , 5.0  ,6.0  ,8.0  ,15.0  ,25.0
0.407     , 0.407,.447,.728,.873 ,.970 ,1.026,1.091 ,1.745,1.745
A0AC,2
9,3
0.0   , 0.01,  0.1 ,    0.5 ,     0.8 ,     1.0 ,     10.0 ,     25.0       ,28.0
-10.0 ,0.0 ,10.0
80.0,    2.00,  0.50,   0.15,      .15,      .15      ,    .51,     1.0      ,1.0
80.0,    2.00,  0.50,   0.15,      .15,      .15      ,    .51,     1.0      ,1.0
80.0,    2.00,  0.50,   0.15,      .15,      .15      ,    .51,     1.0      ,1.0
ISPR,1
4
0    ,            30000,        80000,         350000
390 ,             440,          445,           460
WDOTFMAX,1
4
0.0 ,            8.0 ,          8.01 ,          28
0.0 ,            0.0 ,          0.0   ,         0.0
400.0   , 5500.0
400000 ,150000,250000 ,8000  , 1
0.001 ,0.001 ,0.0 ,0.0 ,      2.0,    10.0 , 0.0
1500.0,1600.0,1400.0, 5.0 , -5.0
390.0 ,15.0
2.00,    .50 , 8.0
0.0292
0        , 1000
0 ,0 ,0
0    ,0    ,0     ,0    ,0
 .1   ,.1   ,   .009   , .1   ,   2.0   ,   .1   , 50.0,900 , 18000
```

```
INPUT DATA FILE "STG2.DAT"/26JUNE88/1530/CONSTANT Q TO ORBIT /PROGRAM"ETO2"
ALL ROCKET SECOND STAGE
ISPA,2
13,3
0.0 , 0.5 ,1.0, 1.5 ,2.0 ,2.5 ,3.0 ,4.0, 5.0 , 5.1  , 8.0 , 15.0 ,  28.0
0.1 ,1.0 ,10.0
4500, 4400,4200,4000,3700,3500,3200,2800,2400, 2400,  2300, 1000,   600
4500, 4400,4200,4000,3700,3500,3200,2800,2400, 2400,  2300, 1000,   500
1000, 1000,1000,1000,1000,1000,1000, 800, 600, 600,   600,  450,    450
PHIMAX,1
12
0.0 ,0.8 ,1.0 ,1.4 ,2.0, 5.0, 5.1, 8.0 ,16.0 ,18.0 ,25.0 ,28.0
0.0 ,0.0 ,0.0 ,0.0 ,0.0, 0.0, 0.0 , 0.0, 0.0 , 0.0 , 0.0 , 0.0
CD0,1
15
0.0,.4 ,.8 ,.9 ,1.0 ,1.1 ,1.2 ,1.5 ,2.0 ,3.0 ,5.0 ,10.0 ,15.0 ,20.0 ,25.0
0.0,.009,.009,.009,.023,.022,.018,.016,.013,.008,.004,.002,.0019,.0017,.0015
DELCD,1
5
0.0,50000.,100000.,150000.,200000.
0.0,.0006 ,.002   ,.001   ,.0020
CLALPHA,1
10
0.0,        1.5 ,2.0 ,3.0 , 4.0 , 5.0  ,6.0  ,8.0  ,15.0  ,25.0
0.043,      .043,.0391,.024, .020, .018,.017,.016, .010 ,.010
K,1
10
0.0,        1.5 ,2.0 ,3.0 , 4.0 , 5.0  ,6.0  ,8.0  ,15.0  ,25.0
0.407     , 0.407,.447,.728,.873 ,.970 ,1.026,1.091 ,1.745,1.745
A0AC,2
9,3
0.0  , 0.01,  0.1 ,    0.5 ,     0.8 ,     1.0 ,     10.0 ,     25.0       ,28.0
-10.0 ,0.0 ,10.0
80.0,   2.00,  0.50,   0.15,     .15,      .15      ,    .51,     1.0      ,1.0
80.0,   2.00,  0.50,   0.15,     .15,      .15      ,    .51,     1.0      ,1.0
80.0,   2.00,  0.50,   0.15,     .15,      .15      ,    .51,     1.0      ,1.0
ISPR,1
4
0    ,         30000,       80000,       350000
390 ,          440,         445,         460
WDOTPMAX,1
4
0.0 ,          8.0 ,        8.01 ,        28
600 ,          600 ,        600  ,        600
100.0  , 1500.0
150000 ,120000, 30000.0 ,27000  , 2
 2.0,   10.0
1000.0,1100.0,900.0, 5.0 , -5.0
2.00,  1.00 , 8.0
0.0292
0       , 1000
0 ,0 ,0
0   ,0    ,0    ,0    ,0
 .1  , .1  ,  .009  , .1  ,  2.0  ,  .1  , 50.0,900 , 18000
```

G:2STG.OUT 01-29-1989 20:17:47

TIME	H0	V0	D0	R	WEIGHT	WDOTF	WDOTA	PHI	WDOTP	RES1
100.00	2167.31	919.81	943.25	10.81	395217.31	27.84	3959.30	0.24	0.00	0.00
200.00	10325.58	923.11	740.38	25.79	392033.19	34.37	3096.59	0.38	0.00	0.00
300.00	25111.56	1474.98	1155.32	42.41	386615.31	87.87	3389.13	0.89	0.00	0.00
400.00	48919.79	2677.66	1373.80	76.95	377626.69	85.09	2914.02	1.00	0.00	0.00
500.00	63469.52	3311.88	1048.32	126.42	370790.59	58.73	2011.35	1.00	0.00	0.00
600.00	63445.13	3982.33	1517.49	185.95	364502.31	48.92	2693.12	0.62	0.00	0.00
700.00	59209.52	4018.83	1892.10	252.07	359204.19	58.11	3345.72	0.59	0.00	0.00
800.00	65138.59	4133.29	1507.68	318.38	353262.19	76.99	2636.56	1.00	0.00	0.00
900.00	73827.55	4732.20	1293.51	391.68	346526.59	62.55	2142.26	1.00	0.00	0.00
1000.00	71043.86	5061.94	1694.84	473.45	340969.69	49.88	2744.32	0.62	0.00	0.00
1100.00	73864.21	5152.98	1531.05	556.93	335650.50	71.63	2460.81	1.00	0.00	0.00
1200.00	81434.77	5694.14	1296.90	646.38	329224.69	58.68	2009.63	1.00	0.00	0.00
1300.00	78853.09	6063.08	1664.94	744.13	323766.00	47.49	2532.66	0.64	0.00	0.00
1400.00	81387.32	6163.81	1523.14	844.11	318647.00	67.23	2302.55	1.00	0.00	0.00
1500.00	88346.38	6666.24	1277.44	949.88	312657.19	54.91	1880.56	1.00	0.00	0.00
1600.00	84914.17	7033.21	1674.76	1063.51	307359.59	46.72	2432.43	0.66	0.00	0.00
1700.00	87016.58	7083.45	1536.60	1179.32	302497.91	62.01	2225.08	0.95	0.00	0.00
1800.00	94202.17	7557.83	1244.34	1299.98	296783.19	51.52	1764.35	1.00	0.00	0.00
1900.00	90241.23	7958.53	1664.04	1428.22	291588.31	45.45	2332.69	0.67	0.00	0.00
2000.00	90952.39	7961.97	1610.27	1559.16	286874.41	45.55	2256.25	0.69	0.00	0.00
2100.00	98501.49	8037.19	1149.67	1690.99	145552.09	0.00	0.00	0.00	48.61	0.00
2200.00	107149.90	9381.72	1043.05	1831.56	129714.20	0.00	0.00	0.00	180.50	0.00
2300.00	114917.40	11303.33	1044.63	2001.82	112933.60	0.00	0.00	0.00	155.76	0.00
2400.00	121578.10	13201.00	1045.10	2203.46	98307.59	0.00	0.00	0.00	136.83	0.00
2500.00	127428.40	15083.61	1045.54	2436.22	85585.84	0.00	0.00	0.00	118.53	0.00
2600.00	132643.59	16948.62	1045.98	2699.81	74562.04	0.00	0.00	0.00	101.71	0.00
2700.00	137355.70	18797.00	1046.30	2993.96	65021.18	0.00	0.00	0.00	88.34	0.00
2800.00	141659.59	20632.69	1046.72	3318.43	56726.34	0.00	0.00	0.00	76.86	0.00
2900.00	145592.50	22444.70	1047.52	3672.94	49525.49	0.00	0.00	0.00	66.85	0.00
3000.00	149177.20	24219.52	1048.93	4056.97	43287.50	0.00	0.00	0.00	57.79	0.00
3064.00	151315.20	25332.92	1049.71	4317.94	39731.11	0.00	0.00	0.00	53.43	0.00
3088.00	152095.20	25757.05	1050.59	4422.21	38426.70	0.00	0.00	0.00	57.71	0.00

TIME	H0	V0	ISP	ISPAVG	ISPEFF	ISPMEAN	THRUST	THRUSTCOM	THRUSTMAX
100.00	2167.31	919.81	4267.90	4366.57	-8.81	2395.35	118834.70	118834.70	493417.81
200.00	10325.58	923.11	4256.79	4326.29	-7.06	1484.12	146289.70	146289.70	384901.41
300.00	25111.56	1474.98	4018.96	4270.74	2206.68	1436.25	353138.81	353138.81	397726.50
400.00	48919.79	2677.66	3340.07	4022.78	987.47	1544.59	284205.00	388000.31	284205.00
500.00	63469.52	3311.88	3031.27	3824.78	1114.59	1435.55	178031.30	297046.31	178031.30
600.00	63445.13	3982.33	2754.19	3661.37	1909.01	1396.94	134734.59	134734.59	216586.50
700.00	59209.52	4018.83	2739.10	3541.64	-1.64	1225.65	159171.50	159171.50	267596.91
800.00	65138.59	4133.29	2691.80	3438.99	954.57	1096.76	207235.59	224654.30	207235.59
900.00	73827.55	4732.20	2455.13	3329.78	885.60	1075.43	153578.00	270389.81	153578.00
1000.00	71043.86	5061.94	2396.17	3242.90	-1.14	1035.78	119516.50	119516.50	192015.00
1100.00	73864.21	5152.98	2393.30	3172.89	778.30	963.33	171421.41	171421.41	171972.00
1200.00	81434.77	5694.14	2375.17	3101.29	804.02	950.75	139378.00	255637.80	139378.00
1300.00	78853.09	6063.08	2361.79	3048.72	-0.93	933.30	112153.40	112153.40	174663.30
1400.00	81387.32	6163.81	2358.61	3005.47	767.07	885.57	158579.70	162870.00	158579.70
1500.00	88346.38	6666.24	2342.01	2960.55	756.51	877.86	128605.40	242200.30	128605.40
1600.00	84914.17	7033.21	2328.56	2924.71	-0.80	867.59	108781.30	108781.30	165390.59
1700.00	87016.58	7083.45	2327.15	2894.98	606.75	828.21	144298.00	144298.00	151200.41
1800.00	94202.17	7557.83	2311.78	2863.13	714.66	821.95	119100.60	229469.70	119100.60
1900.00	90241.23	7958.53	2284.02	2836.14	-0.73	818.16	103804.70	103804.70	158641.00
2000.00	90952.39	7961.97	2284.09	2813.14	-0.71	784.04	104043.70	104043.70	153434.30
2100.00	98501.49	8037.19	446.03	1267.74	0.00	349.79	21682.32	21682.32	267616.69
2200.00	107149.90	9381.72	446.51	1219.60	403.64	349.35	80593.04	80593.04	267905.00
2300.00	114917.40	11303.33	446.94	1174.43	396.78	352.20	69616.70	69616.70	268163.91
2400.00	121578.10	13201.00	447.31	1139.17	389.84	354.16	61206.68	61206.68	268385.91
2500.00	127428.40	15083.61	447.63	1111.18	388.37	355.60	53056.02	53056.02	268580.91
2600.00	132643.59	16948.62	447.92	1088.71	390.65	356.69	45559.97	45559.97	268754.81
2700.00	137355.70	18797.00	448.19	1070.46	389.59	357.57	39592.23	39592.23	268911.81
2800.00	141659.59	20632.69	448.43	1055.43	387.12	358.25	34465.09	34465.09	269055.31
2900.00	145592.50	22444.70	448.64	1042.96	382.40	358.76	29991.06	29991.06	269186.41
3000.00	149177.20	24219.52	448.84	1032.57	376.75	359.11	25940.01	25940.01	269305.91
3064.00	151315.20	25332.92	448.96	1026.80	366.53	359.24	23989.11	23989.11	269377.19
3088.80	152095.20	25757.05	449.01	1024.72	326.06	359.27	25911.53	25911.53	269407.19

TIME	H0	V0	Q0	ALPHA	GAMMA	LIFT	DRAG	ACCEL-G'S	ACCCOM	T/D
100.00	2167.31	919.81	943.25	3.54	3.31	790434.38	95841.65	-0.00	0.00	1.24
200.00	10325.58	923.11	740.38	4.48	6.94	784066.38	98597.94	-0.00	0.00	1.48
300.00	25111.56	1474.98	1155.32	0.00	8.22	0.00	105590.90	0.50	0.50	3.34
400.00	48919.79	2677.66	1373.80	-5.00	4.65	-1039890.00	168502.59	0.23	0.50	1.69
500.00	63469.52	3311.88	1048.32	-5.00	1.31	-643249.00	103528.70	0.18	0.50	1.72
600.00	63445.13	3982.33	1517.49	0.61	-1.18	99953.04	57373.41	0.31	0.23	2.35
700.00	59209.52	4018.83	1892.10	5.00	0.12	1024811.00	157792.30	-0.00	0.00	1.01
800.00	65138.59	4133.29	1507.68	5.00	1.47	806796.13	124345.50	0.21	0.26	1.57
900.00	73827.55	4732.20	1293.51	-5.00	0.22	-650093.00	96261.41	0.16	0.50	1.50
1000.00	71043.86	5061.94	1694.84	5.00	-0.16	829109.50	119989.30	-0.00	0.00	1.00
1100.00	73864.21	5152.98	1531.05	5.00	0.77	745473.38	108474.10	0.16	0.17	1.55
1200.00	81434.77	5694.14	1296.90	-5.00	0.24	-612717.31	90220.30	0.14	0.50	1.54
1300.00	78853.09	6063.08	1664.94	5.00	-0.15	773596.38	112590.30	-0.00	0.00	1.00
1400.00	81387.32	6163.81	1523.14	5.00	0.61	705776.69	103127.90	0.16	0.18	1.54
1500.00	88346.38	6666.24	1277.44	-5.00	0.13	-583474.13	85801.66	0.13	0.50	1.50
1600.00	84914.17	7033.21	1674.76	5.00	-0.19	755966.88	109393.10	-0.00	0.00	0.99
1700.00	87016.58	7083.45	1536.60	5.00	0.52	692738.81	100652.40	0.12	0.13	1.43
1800.00	94202.17	7557.83	1244.34	-5.00	0.15	-553352.38	80989.87	0.13	0.50	1.47
1900.00	90241.23	7958.53	1664.04	5.00	-0.30	728901.88	104898.00	-0.00	0.00	0.99
2000.00	90952.39	7961.97	1610.27	5.00	0.40	705265.69	101686.50	-0.00	0.00	1.02
2100.00	98501.49	8037.19	1149.67	5.00	0.58	137066.80	20130.63	0.00	0.00	1.03
2200.00	107149.90	9381.72	1043.05	0.10	0.53	2381.28	6410.02	0.57	0.57	12.57
2300.00	114917.40	11303.33	1044.63	1.89	0.36	39263.29	7128.71	0.55	0.55	9.77
2400.00	121578.10	13201.00	1045.10	2.39	0.27	44219.36	7434.94	0.55	0.55	8.22
2500.00	127425.40	15083.61	1045.54	2.26	0.21	36739.28	6737.90	0.55	0.54	7.87
2600.00	132643.59	16948.62	1045.98	1.49	0.17	23357.93	5622.90	0.54	0.54	8.10
2700.00	137355.70	18797.00	1046.30	0.97	0.14	15198.83	5024.11	0.54	0.54	7.88
2800.00	141659.59	20632.69	1046.72	0.44	0.11	6631.42	4584.90	0.53	0.53	7.52
2900.00	145592.50	22444.70	1047.52	-0.14	0.10	-2140.31	4318.83	0.53	0.52	6.94
3000.00	149177.20	24219.52	1048.93	0.20	0.08	3120.03	4120.82	0.51	0.51	6.29
3064.00	151315.20	25332.92	1049.71	0.87	0.07	13643.94	4274.48	0.50	0.50	5.61
3098.80	152095.20	25757.05	1050.59	3.34	0.07	52615.54	7138.62	0.49	0.49	3.63

4.0 LISTING OF PROGRAM ETO

In the following program listing, the leftmost column of numbers are sequential program line numbers that were generated by the file cross referencing code used to generate the program listing. They are not part of the program. These line numbers are used with the cross referenced listing of program variables in Section 5.0.

```
' 1615/14JAN89    ETO13.BAS (Date of last update)
'
' ********************************************************************
' *                                                                  *
' *                          ETO13.BAS                               *
' *                DEVELOPMENT VERSION OF "ETO.BAS"                  *
' ********************************************************************
'                           WRITTEN BY:
'                        JOHN L. LEINGANG
'                        WAYNE A. DONALDSON
'                        KENNETH A. WATSON
'                      2LT LOUIS R. CARREIRO
'
'                           AFWAL/POPA
'                           WPAFB,OHIO
'     Written during period January 1986 to February 1987
'
'  This is a 2-DOF trajectory program for calculating the earth-to-orbit
'  performance of single stage or two stage horizontal takeoff vehicles.
'  The program assumes a spherical non-rotating earth.A 2nd order integ-
'  ration is used for the equations governing weight,velocity,flight path
'  angle,altitude,and range.
'
'  Rocket-only,airbreathing-only,and simultaneous rocket and airbreath-
'  ing propulsion may be used in any part of the trajectory.
'
'  Trajectory options are:
'        *  Cruise at constant Mach number and altitude
'        *  Cruise at constant Mach number and constant L/D
'        *  Climb to orbit at constant Q
'        *  Climb to orbit along a velocity-biased Q path
'        *  Climb to orbit ballistically,beginning at a commanded velocity
'
' Algorithms for table loading and lookup developed by Ken Watson.
' Algorithm for constant Q0 path developed by Wayne Donaldson.
'
' **********************************************************************
'
       CLS
       WIDTH LPRINT 80
           DIM I1ST(50), J1ST(50), T$(50), IO(50), AT(1000)
           DIM TG(3), GA(3)
           DIM DELW(4), DELV(4), DELGAM(4), DELH(4), DELR(4)
           DIM RH0B(10), GLMB(10), HB(10), TMB(10)
     INPUT "IS THIS A SINGLE-STAGE(1),OR A 2-STAGE VEHICLE (2)? ", NSTAGES
     IF NSTAGES = 2 GOTO 460
     INPUT "SINGLE STAGE FILE NAME"; IFN$
                                         GOTO 470
460  INPUT "FIRST STAGE FILE NAME"; IFN$
     INPUT "SECOND STAGE FILE NAME"; IFN2$
'
470        OPEN "I", #1, IFN$: IL = 0: IFILE = 1
           GOTO 480
'
480  LINE INPUT #1, TITL1$
     LINE INPUT #1, TITL2$
     '
     GOSUB 16000:      'Input airbreather Isp vs. Mach and PHI as table #1
     GOSUB 16000:      'Input airbreather PHImax vs. Mach as table #2
     GOSUB 16000:      'Input Cdo vs.Mach as table #3
     GOSUB 16000:      'Input delta Cd vs. Alt. as table #4
     GOSUB 16000:      'Input Clalpha vs. Mach as table #5
     GOSUB 16000:      'Input K vs. Mach as table #6
     GOSUB 16000:      'Input airbreather Ao/Ac vs. Mach and alpha as table #7
     GOSUB 16000:      'Input rocket Isp vs. Altitude as table #8
     GOSUB 16000:      'Input rocket WDOTPMAX vs.Mach as table #9

      INPUT "Do you want a listing of tabular input data(Y/N)"; WTIN$
      IF WTIN$ = "Y" OR WTIN$ = "y" THEN 630 ELSE 640
630   FOR ITAB = 1 TO 9: GOSUB 16230: NEXT ITAB
640 '
```

```
        INPUT "Do you want printer output during run(Y/N)"; PRCOM$
        IF PRCOM$ = "Y" OR PRCOM$ = "y" THEN PRCOM = 1 ELSE PRCOM = 0
        IF PRCOM = 1 THEN GOTO 680 ELSE GOTO 700
680 ''' LPRINT CHR$(27); "B"; CHR$(3); : 'Select condensed print (SR15 printer)
        LPRINT CHR$(27); CHR$(33); CHR$(4); : 'Select condensed print(Epson printer)
        GOTO 710
700 ''' LPRINT CHR$(27); "B"; CHR$(1); : 'Select standard print (SR15 printer)
      ''LPRINT CHR$(27); CHR$(33); CHR$(0); : 'Select standard print (Epson printer)
      ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
710   INPUT #1, AC, AREF
      INPUT #1, WLAUNCH, WFUEL, WFINAL, VFINAL, STAGE
      INPUT #1, M0, H0, GAMMA, ALPHA, DT, DELPRINT, TIMEX
      INPUT #1, Q0COMD, Q0MAX, Q0MIN, ALPHAMAX, ALPHAMIN
      INPUT #1, VTAKEOFF, ALPHA2MAX
      INPUT #1, LOADFAC, ACCCOMD, GAMMAMAX
      INPUT #1, FASTOIC
      INPUT #1, V0TEMP, Q0FINAL
      INPUT #1, V0CRUISE, SWITCH4, GAMMA4MAX
      INPUT #1, FLAG5, V05, ALPHA5MAX, GAMMA5, ACCCOMD5
      INPUT #1, A1, A2, B1, C1, C2, D1, DELQ, V02, H02
      '
      PRINT
      PRINT "1ST/SINGLE STAGE INPUT FILE HAS BEEN READ."
      PRINT "LAST ITEM IN INPUT FILE IS H02,VALUE READ FROM FILE IS"; H02
      PRINT "IF THIS VALUE IS INCORRECT,INPUT FILE IS IN ERROR."
      PRINT
      INPUT "Do you want a plotter/printer file created? (Y/N)"; PLOT$
      IF PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 1020
      INPUT "Enter the plotter/printer file name (DRIVE:NAME.EXT)    ", PFN$
      OPEN "O", #2, PFN$
      VARNUM = 25
      WRITE #2, VARNUM
      PRINT
      PRINT
      PRINT "At the prompt below,enter a value for FREQ,the interval of plotter/"
      PRINT "printer output."
      PRINT "A value of 1 for FREQ means output at each time increment  "
      PRINT "A value of 2 means every other time increment,etc. As a guide to "
      PRINT "the amount of disk space required,use the following:"
      PRINT
      PRINT "  BYTES REQUIRED = (200) X  (FLIGHT TIME)"
      PRINT "                   -------------------------------"
      PRINT "                   (INTEGRATION INTERVAL) X (FREQ)"
      PRINT
      PRINT " Very large files exceeding 200K may be generated.Exit the "
      PRINT " program by pressing CTRL C if disk space is insufficient,"
       PRINT " otherwise enter value for FREQ below:"
       INPUT "                                        ", FREQ
1020   STARTTIME$ = TIME$: START = TIMER
           IF EOF(1) THEN GOTO 1022
1022       CLOSE #1
      '-------------------------------------------------------------
      '    Use this area to the line below as a "scratch" space to
      '    temporarily overide values read from the 1st/single stage
      '    input file.


''''' DT = .1
''''  DELPRINT = 1

    '
      ' -------------------------------------------------------------------
      CLS :                    'Clears screen
       '
      '------------------------ EVALUATE CONSTANTS -------------------
      '
      GC = 32.174
      MEARTH = (5.98E+27) / 454:                        'Earth mass
      REARTH = (6.38E+08) / (2.54 * 12):                'Earth radius
      BIGG = (6.67E-11) * (1 / 4.4482) * (.4536 / .3048) ^ 2: 'Universal gravitation const

      FC = (WEIGHT / GC) * (V0) ^ 2 / (H0 + REARTH): 'Centrifugal force
```

```
      ANG = 57.3:  'Degrees per radian
      '
      '
      '------------------- INITIALIZE THE PROBLEM --------------------
      '
      IF TIMEX = 0! THEN WEIGHT = WLAUNCH
      COUNT = 0
       PRCOUNT = 0
       I = 0
       '------Load data for atmosphere routine -------------
        FOR J1 = 1 TO 10
       '
        READ HB(J1): READ RH0B(J1): READ TMB(J1): READ GLMB(J1)
         NEXT J1
      '
2540    DATA 0,2.3769E-3,518.7,-3.5662E-3
        DATA 36089,7.0612E-4,390,0
        DATA 65617,1.7082E-4,390,5.4864E-4
        DATA 104986,2.5661E-5,411.6,1.5362E-3
        DATA 154199,2.7698E-6,487.2,0
       DATA 170604,1.4735E-6,487.2,-1.0973E-3
       DATA 200131,4.8719E-7,454.8,-2.1946E-3
       DATA 259186,3.8826E-8,325.2,0
       DATA 291153,6.1508E-9,325.2,1.6953E-3
       DATA 323003,9.6511E-10,374.2,2.8343E-3
            RESTORE 2540
            '
       '$PAGE
```

```
      '***************************************************************************
      '                     MAIN PROGRAM ROUTINE
      '***************************************************************************
      '
      '---------Get current values from input data:
       ITAB = 3: XARG = M0: GOSUB 16400: CD0 = VALI: 'Look up Cd0
       ITAB = 4: XARG = H0: GOSUB 16400: DELCD = VALI: 'Look up delCd friction
       ITAB = 5: XARG = M0: GOSUB 16400: CLALPHA = VALI: 'Look up Clalpha
       ITAB = 6: XARG = M0:                GOSUB 16400: K = VALI: 'LOOK UP K
       ITAB = 7: XARG = M0: YARG = ALPHA: GOSUB 16400: A0AC = VALI: 'LOOK UP A0AC

      '------------- CALCULATE CURRENT LIFT AND DRAG -------------------
      '
      CL = ALPHA * CLALPHA
      LIFT = CL * Q0 * AREF
      CD = CD0 + K * CL ^ 2 + DELCD
      DRAG = CD * Q0 * AREF


       FOR I = 0 TO 1000000!
''''''
3100   A$ = INKEY$: IF A$ = CHR$(3) THEN GOTO 13260:  'Intercepts CTRL-C entered
       '                           from keyboard to halt execution and exit program
       Q$ = INKEY$: IF Q$ = CHR$(19) THEN GOTO 3108 ELSE GOTO 3118: 'Intercepts
       '                         CTRL-S to halt execution until space bar is pressed
3108 '
3110   Q$ = INKEY$
       IF Q$ <> CHR$(32) THEN GOTO 3110: 'Resumes if space bar is pressed
      ''
3118 '
       GOSUB 13500: ' Go to atmosphere subroutine and return here

      GA = BIGG * MEARTH / (H0 + REARTH) ^ 2 * GC: 'Gravitational acceleration
      '----------Reduce integration interval near end of flight
      IF WEIGHT - WFINAL < .01 * WFINAL THEN DT = .1
      IF V0 < 24000 THEN GOTO 3700
      DV = CINT((26200 - V0) / 1000)
      DT = DV
      IF DT = 0 THEN DT = .1
3700 '
       GOTO 4900
4900 '

      '-------------------- FLIGHT PHASE COMMANDS --------------------
      '
       PHASE = 1
      '      Takeoff Roll at zero alpha until VTAKEOFF is reached
          IF V0 >= VTAKEOFF OR TIMEX > TTAKEOFF THEN 6500: 'Test to see if
       '                                                    Phase 1 is completed
          ALPHA = 0!: GAMMA = 0!: LIFT = WEIGHT: H0 = 0!
           ACCCOM = ACCCOMD
          GOSUB 14000:    'Go to thrust subroutine and return here
          ALPHA = 0!: GAMMA = 0!: LIFT = WEIGHT: H0 = 0!
       '
       '
          TTAKEOFF = TTAKEOFF + DT: A$ = "PHASE 1,TAKEOFF ROLL **************"
          GOSUB 15800:  ' Go to integration subroutine then return here
          GOTO 12000:   ' Go to runtime output section
6500 '---------------------------------------------------------------------
       PHASE = 2: 'Commanded climb at constant load factor. Required ALPHA is
      '      calculated,but is limited to ALPHAMAX. Maximum velocity allowed
      '      is V02 fps. ALPHA is commanded to zero above  H02  ft. This phase
      '      is maintained until V02 fps,H02 ft,and 0.95*Q0COMD are reached.
      '
           IF TIMEX > TTAKEOFF + TQ0 + DT THEN 6860:     'Test if Phase 2 is
      '                                                   completed
          IF V0 > V02 AND H0 > H02 AND Q0 > .95 * Q0COMD THEN GOTO 6860
              ALPHA = WEIGHT * LOADFAC / (CLALPHA * Q0 * AREF)
           IF ALPHA > ALPHA2MAX THEN ALPHA = ALPHA2MAX
              ACCCOM = ACCCOMD
          IF V0 > V02 AND H0 < H02 THEN ACCCOM = 0
```

```
                IF H0 > H02 THEN ALPHA = 0
                 TQ0 = TQ0 + DT
                 B$ = "PHASE 2,CLIMB AT CONSTANT L/W****************"
               GOSUB 14000:    'Go to thrust subroutine and return here
                GOSUB 15800:  'Go to integration subroutine then return here
               GOTO 12000:     'Go to runtime output section


6860 '-----------------------------------------------------

       PHASE = 3: 'Commanded climb at constant Q0. An angle of attack (ALPHA)
      '     is computed that will correct the flight path angle (GAMMA) to
      '     seek the commanded Q0COM.
      '     This is GUID13 version.  29SEPT86. THIS ALGORITHM DEVELOPED BY
      '     WAYNE DONALDSON
      '
      '
           Q0COM = Q0COMD
           IF V0 < V0TEMP THEN GOTO 7100
                   IF FLAG5 > 0 AND TSET5 = 1 THEN GOTO 8300
           Q0COM = Q0COMD - (Q0COMD - Q0FINAL) * (V0 - V0TEMP) / (26200 - V0TEMP)
      '                                Q0COMD is reduced for V0 > V0TEMP
7100       IF Q0 < Q0COM THEN GOTO 7320
              IF Q0 < QOLD THEN GOTO 7220
      '
                 RATEDQ = (Q0 - QOLD) / DT
7180             ALPHA = ALPHA + A1 + RATEDQ * A2
                 GOTO 7500
7220             RATEDELQ = (Q0 - Q0COM) / DT
                 IF Q0 - Q0COM > DELQ THEN RATEDQ = (QOLD - Q0) / DT
                 IF Q0 - Q0COM > DELQ THEN GOTO 7180
                 ALPHA = ALPHA - RATEDELQ * B1
                 GOTO 7500
7320          IF Q0 > QOLD THEN GOTO 7420
      '
                 RATEDQ = (Q0 - QOLD) / DT
7380             ALPHA = ALPHA - C1 + RATEDQ * C2
                 GOTO 7500
7420             RATEDELQ = (Q0 - Q0COM) / DT
                 IF Q0COM - Q0 > DELQ THEN RATEDQ = (QOLD - Q0) / DT
                 IF Q0COM - Q0 > DELQ THEN GOTO 7380
                 ALPHA = ALPHA + RATEDELQ * D1
7500    QOLD = Q0
             IF ALPHA > ALPHAMAX THEN ALPHA = ALPHAMAX
             IF ALPHA < ALPHAMIN THEN ALPHA = ALPHAMIN
       '      ------------------------------------------------
      '                 CALCULATE ACCELERATION COMMAND FOR PHASE 3
      '                 -----------------------------------------
      '
             IF Q0 <= Q0COM GOTO 7660 ELSE GOTO 7780
7660 ''''''''For Q0 < Q0COM, maximum available acceleration is commanded
      '
             ACCCOM = ACCCOMD
             GOTO 7980
       '
      '
7780 ''''''''For Q0 > Q0MAX, acceleration is reduced  linearly to zero for Q0
      '                  between Q0COM and Q0MAX and is held to zero when
      '                  Q0MAX is exceeded
      '
             ACCPERQ = ACCCOMD / (Q0COM - Q0MAX)
             ACCCOM = ACCCOMD - ACCPERQ * (Q0COM - Q0)
      '
               IF Q0 > Q0MAX THEN ACCCOM = 0
               GOTO 7980
       '
7980 ''''''''''
      '
                     Q0OLD = Q0
                     RH0OLD = RH0
                     H0OLD = H0
              IF V0CRUISE > 0 AND V0 >= V0CRUISE THEN GOTO 8180
```

```
            IF FLAG5 > 0 AND V0 > V05 THEN GOTO 8300
          C$ = "PHASE 3,CLIMB AT COMMANDED Q****************"
   GOSUB 14000:      'Go to THRUST subroutine and return here
   GOSUB 15800:      'Goto integration subroutine and return here
   GOTO 12000:       'Go to runtime output section

  ' ----------------------------------------------------------------
8180   PHASE = 4: 'Cruise at commanded V0CRUISE and Q0COM
                 ACCCOM = 0
   IF SWITCH4 = 0 THEN GOTO 8260
  IF FLAGCRUISE = 1 THEN GOTO 8250
  ''''
  FOR ALPHA = 0 TO ALPHAMAX STEP .1
   ITAB = 3: XARG = M0: GOSUB 16400: CD0 = VALI: 'Look up Cd0
   ITAB = 4: XARG = H0: GOSUB 16400: DELCD = VALI: 'Look up delCd friction
   ITAB = 5: XARG = M0: GOSUB 16400: CLALPHA = VALI: 'Look up Clalpha
   ITAB = 6: XARG = M0: GOSUB 16400: K = VALI: 'LOOK UP K
  CL = ALPHA * CLALPHA
  CD = CD0 + K * CL ^ 2 + DELCD
  LOVRD = CL / CD: PRINT ALPHA; LOVRD
  IF LOVRD < LOVRDOLD THEN GOTO 8225
  LOVRDOLD = LOVRD
  NEXT ALPHA
  '
8225 '------------------------------------
  ALPHACRUISE = ALPHA - .1
  '
  FLAGCRUISE = 1
8250 ALPHA = ALPHACRUISE
  IF GAMMA > GAMMA4MAX THEN ALPHA = 0!
  IF LIFT < WEIGHT THEN ALPHA = 0!
  IF GAMMA < 0! THEN ALPHA = ALPHAMAX

8260 '------------------------------------------------------------------
          D$ = "PHASE 4,CRUISE AT CONSTANT ALTITUDE ************************"
   IF SWITCH4 = 1 THEN D$ = "PHASE 4,CRUISE AT CONSTANT L/D **************"
       GOSUB 14000: 'Go to thrust subroutine and return here
       GOSUB 15800: 'Goto integration subroutine and return here
       GOTO 12000:  'Go to runtime output section
8300 '------------------------------------------------------------------
    PHASE = 5: ' At V05 begin pullup to flight path angle GAMMA5 then
  '               continue ballistically at commanded acceleration ACCCOMD5
  '
           ACCCOM = ACCCOMD5
          ALPHA = ALPHA5MAX: ALPHAMAX = ALPHA5MAX
          IF TSET5 = 1 THEN ALPHA = 0
          IF TSET5 = 1 THEN GOTO 8500
          IF GAMMA >= GAMMA5 THEN TSET5 = 1 ELSE TSET5 = 0
  '
8500      E$ = "PHASE 5,PULLUP AND BALLISTIC ASCENT"
   GOSUB 14000: 'Go to thrust subroutine and return here
        GOSUB 15800: 'Go to integration subroutine
        GOTO 12000:  'Go to runtime output section
   '
12000 '
   '--------------- RUNTIME OUTPUT SECTION -----------------------
   '
   '
   ISPEFF = WEIGHT * (V0 - V00) / (GC * (WEIGHT0 - WEIGHT))
   ITCUM = ITCUM + ISPEFF * (WEIGHT0 - WEIGHT)
   ISPMEAN = ITCUM / (WLAUNCH - WEIGHT)
   '
   ITSUM = ITSUM + ISP * (WEIGHT0 - WEIGHT)
   ISPAVG = ITSUM / (WLAUNCH - WEIGHT)
   ACC = (V0 - V00) / (DT * GA)
   '
   '
   'First test for weight or velocity stopping condition:
         VORBIT = REARTH * SQR(GC / (REARTH + H0))
         IF V0 >= VORBIT GOTO 13020
         IF WEIGHT <= WFINAL GOTO 13140
        IF V0 >= VFINAL AND STAGE = 1 THEN GOTO 13190
```

```
        '
        '
          COUNT = COUNT + 1
          IF COUNT <= (DELPRINT - 1) THEN GOTO 12980
        CLS
        IF PHASE = 1 THEN X$ = A$ ELSE IF PHASE = 2 THEN X$ = B$ ELSE IF PHASE = 3 THEN X$ = C$ ELSE IF PHASE = 4 THEN X$ = D$ ELSE IF PHASE = 5 THEN X$ = E$
       PRINT USING "T=####.#  \                                        \"; TIMEX; X$
       PRINT USING "V0=#####.# M0=##.#  ALPHA=###.##          WDOTF=####.##      RES1=#####.#"; V0; M0; ALPHA; WDOTF; RES1
       PRINT USING "H0=######.#          GAMMA=###.##          WDOTA=####.##      WDOTP=####.#"; H0; GAMMA; WDOTA; WDOTP
       PRINT USING " R=######.##              Q0=####.##            PHI=##.### "; RANGE / 6076; Q0; PHI
       PRINT USING "ISP=#####.##     ALPHAMAX=###.##               WT=#######.#      L/W =###.###"; ISP; ALPHAMAX; WEIGHT; LIFT / WEIGHT
       PRINT USING "ISPEFF=####.#    ALPHAMIN=###.##               FC=#######.#"; ISPEFF; ALPHAMIN; FC
       PRINT USING "ISPMEAN=####.#    ACCCOM =###.##          LIFT=#######.#      L/D =###.###"; ISPMEAN; ACCCOM; LIFT; LIFT / DRAG
       PRINT USING "ISPAVG=#####.#   ACCEL-G'S=###.##         DRAG=#######.#"; ISPAVG; ACC; DRAG
       PRINT USING "ISPA/B=#####.#                            THRUST=#######.#      T/D =###.###"; ISPA; THRUST; THRUST / DRAG
       PRINT USING "ISPROC=#####.#                       THRUSTCOM=#######.#"; ISPR; THRUSTCOM
       PRINT USING "                                     THRUSTMAX=#######.#      T/W =###.###"; THRUSTMAX; THRUST / WEIGHT
        '
        COUNT = 0!
        '
        '
        IF PRCOM <> 1 GOTO 12980
        '
        IF PRCOUNT < 1 GOTO 12660
        IF 2 <= PRCOUNT <= 14 GOTO 12780
12660  LPRINT CHR$(12)
       IF PRCOUNT < 1 THEN LPRINT TAB(5); TITL1$, TAB(5); TITL2$, DATE$, TIME$
       WIDTH LPRINT 136
       LPRINT TAB(5); "TIME"; TAB(15); "V0"; TAB(25); "H0"; TAB(35); "R"; TAB(45); "ISP"; TAB(55); "ISPEFF"; TAB(65); "ISPMEAN"; TAB(75); "WDOTF"; TAB(85); "WDOTA"; TAB(97); "PHI"; TAB(105); "THRUST"; TAB(112); "THRUSTCOM"; TAB(122); "THRUSTMAX"
       LPRINT TAB(15); "Q0"; TAB(25); "ALPHA"; TAB(35); "GAMMA"; TAB(45); "WEIGHT"; TAB(55); "LIFT"; TAB(65); "DRAG"; TAB(75); "WDOTP"; TAB(85); "RES1"; TAB(94); "ISPAVG"; TAB(102); "ACCEL-G'S"; TAB(115); "ACCCOM"; TAB(128); "T/D"
       '
12780 LPRINT
        PRCOUNT = PRCOUNT + 1
       WIDTH LPRINT 136
       '
       LPRINT USING "#######.##"; TIMEX; V0; H0; RANGE / 6076; ISP; ISPEFF; ISPMEAN; WDOTF; WDOTA; PHI; THRUST; THRUSTCOM; THRUSTMAX
       LPRINT USING "#######.##"; TIMEX; Q0; ALPHA; GAMMA; WEIGHT; LIFT; DRAG; WDOTP; RES1; ISPAVG; ACC; ACCCOM; THRUST / DRAG
       WIDTH LPRINT 80
       '
       IF PRCOUNT >= 16 THEN PRCOUNT = 0
       '
12980  NEXT I
       '
       '
```

```
13020 '-------------- TEST FOR STOPPING CONDITIONS --------------------
      '
      '
          PRINT USING "REACHED ORBITAL VELOCITY=######.# FPS AT TIME=#####.#SECS. ,WEIGHT= ######.# LBS."; V0; TIMEX; WEIGHT
          IF PRCOM = 1 THEN LPRINT USING "REACHED ORBITAL VELOCITY=######.# FPS AT TIME=#####.#SECS.,WEIGHT= ######.# LBS."; V0; TIMEX; WEIGHT
             GOTO 13220
13140     PRINT USING "FUEL EXHAUSTED AT TIME= #####.#SECS.,WEIGHT= ######.#,VELOCITY=######.# FPS."; TIMEX; WEIGHT; V0
          IF PRCOM = 1 THEN LPRINT USING "FUEL EXHAUSTED AT TIME= #####.#SECS.,WEIGHT= ######.#,VELOCITY=######.#FPS."; TIMEX; WEIGHT; V0
             GOTO 13220
13190     PRINT USING "REACHED 1ST STAGE VELOCITY=######.# FPS AT TIME=#####.#SECS. ,WEIGHT= ######.# LBS."; V0; TIMEX; WEIGHT
          IF PRCOM = 1 THEN LPRINT USING "REACHED 1ST STAGE VELOCITY=######.# FPS AT TIME=#####.#SECS.,WEIGHT= ######.# LBS."; V0; TIMEX; WEIGHT
             GOTO 13220
       '-------------------------------
13220   IF NSTAGES = 1 THEN GOTO 13250
13240   IF NSTAGES = 2 AND STAGE = 1 THEN GOSUB 18000: 'Reads 2nd stage data
              IF TIMEX = TIMEX2 THEN GOTO 3100
13250   IF PRCOM = 0 AND PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 13300
      ''
13260 ''LPRINT CHR$(27); "B"; CHR$(1); : 'Select standard print (SR15 printer)
      LPRINT CHR$(27); CHR$(33); CHR$(0); : 'Select standard print (Epson printer)
        WIDTH LPRINT 80
      IF PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 13300
      INPUT "Do you want printer output of run(Y/N)"; PLOT$
      IF PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 13300
      CLOSE 2
      IF PLOT$ = "Y" OR PLOT$ = "y" THEN GOSUB 20000:  'Goes to printer output subroutine
.
13300 '''
      PRINT TIME$
      PRINT STARTTIME$: FINISH = TIMER
      PRINT "PROGRAM TOOK"; FINISH - START;
      PRINT "SECONDS"
        END
        '
        '-------------End of main routine ----------------------------
        '
        '
        '
        '
      '$PAGE
```

```
      '*************************************************************
      '*************************************************************
      '
      '                       SUBROUTINES
      '
      '*************************************************************
      '*************************************************************
      '
13500 '*************************************************************
      '         SUBROUTINE TO DEFINE ATMOSPHERIC CONDITIONS
      '                          USING
      '               U.S. STANDARD ATMOSPHERE, 1962
      '                      4FEB87 VERSION
      ' ************************************************************
      '
      '------------------------- DATA ---------------------
      'FOR J1 = 1 TO 10
      'READ HB(J1):READ RH0B(J1):READ TMB(J1):READ GLMB(J1)
      ' NEXT J1
      '
13540 'DATA 0,2.3769E-3,518.7,-3.5662E-3
      'DATA 36089,7.0612E-4,390,0
      'DATA 65617,1.7082E-4,390,5.4864E-4
      'DATA 104986,2.5661E-5,411.6,1.5362E-3
      'DATA 154199,2.7698E-6,487.2,0
      'DATA 170604,1.4735E-6,487.2,-1.0973E-3
      'DATA 200131,4.8719E-7,454.8,-2.1946E-3
      'DATA 259186,3.8826E-8,325.2,0
      'DATA 291153,6.1508E-9,325.2,1.6953E-3
      'DATA 323003,9.6511E-10,374.2,2.8343E-3
      '     RESTORE 13540
      ' - - - - - - - - - SET CONSTANTS - - - - - - - - - -
             N1 = 10
             GC = 32.17405
             RE = 2.0899E+07:    ' radius of earth
             CPCV = 1.4:         ' ratio of specific heats
      ' - - - - - CONVERT TO GEOPOTENTIAL ALTITUDE - - - - -
       H = (RE * H0) / (RE + H0): ' geopotential altitude
       '- - - - - -   FINDS ALTITUDE BASE POINT - - - - - - -
       FOR I1 = 2 TO N1
       HDEL = H - HB(I1)
       IF HDEL <= 0 GOTO 13740
       NEXT I1
      '
      '
13740 ' - - - - - - - -  CALCULATE TEMPERATURE AND DENSITY - - - -
       IF GLMB(I1 - 1) <> 0 GOTO 13840: 'checks to see if temp changes with alt.
       R0 = RH0B(I1 - 1) * EXP(-(H - HB(I1 - 1)) * GC / (1716.483 * TMB(I1 - 1)))
       T0 = TMB(I1 - 1): ' temp doesnt change with alt
       GOTO 13880
13840  T0 = TMB(I1 - 1) + GLMB(I1 - 1) * (H - HB(I1 - 1)): 'temp changes with alt
       R0 = RH0B(I1 - 1) * EXP(-(1! + GC / (1716.483 * GLMB(I1 - 1))) * LOG(T0 / TMB(I1 -
 1)))
13880  '- - - - - -   AMBIENT CONDITIONS - - - - - - - - - - -
       RH0 = R0 * GC:      ' converts density to lbm/cu.ft
       P0 = RH0 * T0 * 53.28 / 144: 'pressure in psia
      '
       C0 = 49.01 * SQR(T0): ' ft/sec
       V0 = M0 * C0
       Q0 = RH0 * (V0 ^ 2) / (2 * GC): ' lbf/sqft
       FC = (WEIGHT / GC) * (V0 ^ 2) / (H0 + REARTH)
       G0 = (P0 * V0) / (53.28 * T0): ' lbm/(sec*sqin)
      RETURN
        '
       '$PAGE
```

```
14000 '************************************************************************
      '                      THRUST CALCULATION SUBROUTINE
      '************************************************************************
        'First update lift and drag:
      '
       ITAB = 3: XARG = M0: GOSUB 16400: CD0 = VALI: 'Look up Cd0
       ITAB = 4: XARG = H0: GOSUB 16400: DELCD = VALI: 'Look up delCd friction
       ITAB = 5: XARG = M0: GOSUB 16400: CLALPHA = VALI: 'Look up Clalpha
       ITAB = 6: XARG = M0: GOSUB 16400: K = VALI: 'LOOK UP K
      CL = ALPHA * CLALPHA
      LIFT = CL * Q0 * AREF
      CD = CD0 + K * CL ^ 2 + DELCD
      DRAG = CD * Q0 * AREF
      '
      '------------------------------------------------------------------
       '  Calculate commanded THRUST value
      '
        THRUSTCOM = (WEIGHT / (GC * COS(ALPHA / ANG))) * (ACCCOM * GA + GA * SIN(GAMMA / ANG)) + DRAG / (COS(ALPHA / ANG))
      ' -----------------------------------------------------------------
      ''AIRBREATHER THRUST-----------------------------------------------
       ITAB = 7: XARG = M0: YARG = ALPHA: GOSUB 16400: A0AC = VALI: 'Look up A0/AC
       ITAB = 2: XARG = M0: GOSUB 16400: PHIMAX = VALI: 'Look up PHImax
      PHI = PHIMAX
       ITAB = 1: XARG = M0: YARG = PHI: GOSUB 16400: ISPA = VALI: 'Look up airbreathing Isp

         A0 = A0AC * AC
                      IF PHI = 0 THEN A0 = 0
         WDOTA = G0 * A0 * 144
         FAMAX = FASTOIC * PHI
      '
         WDOTFMAX = FAMAX * WDOTA
         THRUSTAMAX = ISPA * WDOTFMAX
      ''ROCKET THRUST---------------------------------------------------
      ''
        ITAB = 8: XARG = H0: GOSUB 16400: ISPR = VALI: 'Look up rocket Isp
        ITAB = 9: XARG = M0: GOSUB 16400: WDOTPMAX = VALI: 'Rocket propellant flow

             THRUSTRMAX = ISPR * WDOTPMAX
             THRUSTMAX = THRUSTAMAX + THRUSTRMAX
             THRUST = THRUSTMAX
             WDOTF = WDOTFMAX
             WDOTP = WDOTPMAX
             WDOT = WDOTFMAX + WDOTPMAX
             ISP = THRUST / WDOT
      '
             IF THRUST < THRUSTCOM GOTO 15660: '---------------------------
      '
      ' ------------------------------------------------------------------
      ''' For THRUSTMAX > THRUSTCOM reduce rocket to zero if needed and
       '  reduce airbreathing thrust as needed to obtain THRUSTCOM
      '
         IF WDOTPMAX = 0 THEN GOTO 15260: '-------------------------
      '
          DELTHRUST = THRUSTMAX - THRUSTCOM
          IF DELTHRUST > THRUSTRMAX THEN WDOTP = 0
          IF WDOTP = 0 THEN GOTO 15260: '-----------------------------
      ''''Throttle rocket to reduce thrust
          THRUSTR = THRUSTCOM - THRUSTAMAX
          WDOTP = THRUSTR / ISPR
          WDOTF = WDOTFMAX
          WDOT = WDOTF + WDOTP
          THRUST = THRUSTAMAX + THRUSTR
          ISP = THRUST / WDOT
         GOTO 15660: '----------------------------------------------
15260      IF THRUST = 0! THEN THRUST = 1!
           IF A0AC = 0 OR PHIMAX = 0 OR ISPA = 0 THEN GOTO 15660: '-------
              PHI = PHIMAX
      ''
15340 '
```

```
                ITAB = 1: XARG = M0: YARG = PHI: GOSUB 16400: ISPA = VALI: 'Look up Isp
             '
                THRUST = ISPA * WDOTF
                WDOTF = WDOTF * THRUSTCOM / THRUST
       '
                 PHI = (WDOTF / WDOTA) / FASTOIC
       '
                 IF ABS(THRUST - THRUSTCOM) < 200 GOTO 15560
         '
                GOTO 15340: ' Returns to try new PHI value
15560 '
                WDOTF = PHI * FASTOIC * WDOTA
                THRUST = ISPA * WDOTF
                WDOT = WDOTF
                ISP = ISPA
15660 '
          IF WDOTF = 0 THEN ISPA = 0
          IF WDOTP = 0 THEN ISPR = 0
       '
        RETURN
        '
        '
       '$PAGE
```

```
15800 '**************************************************************************
      '                     INTEGRATION SUBROUTINE
      '            2ND ORDER R-K INTEGRATION (HEUN'S METHOD)
      '**************************************************************************
      IF PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 15920
      ' --------------------CREATE/ADD TO PLOTTER FILE--------------------
      IF FREQC < FREQ THEN GOTO 15910
      FREQC = 0
      WRITE #2, TIMEX, V0, H0, RANGE / 6076, ISP, ISPEFF, ISPMEAN, WDOTF, WDOTA, PHI,
THRUST, THRUSTCOM, THRUSTMAX, Q0, ALPHA, GAMMA, WEIGHT, LIFT, DRAG, WDOTP, RES1, ISPAVG,
ACC, ACCCOM, THRUST / DRAG
15910 FREQC = FREQC + 1
      '-----------------------------------------------------------------
15920  '
      '
      '-------------------Set "old" values of integration variables------------
      TIMEX0 = TIMEX
      WEIGHT0 = WEIGHT
      V00 = V0
      GAMMA0 = GAMMA
      H00 = H0
      RANGE0 = RANGE
      '
      '-------------------Start integration----------------------------------
     FOR N = 1 TO 2
     '
     '
     '--------------------Weight increment----------------------------------
      DELW(N) = (-WDOT) * DT / 1
     '--------------------Velocity increment--------------------------------
     IF TIMEX <= DT THEN WEIGHT = WLAUNCH
      DELV(N) = (GC / WEIGHT) * ((THRUST * COS(ALPHA / ANG) - DRAG - WEIGHT * (GA / GC) *
 SIN(GAMMA / ANG))) * DT / 1
    '--------------------Flight path angle increment-----------------------
     IF V0 < DT THEN GOTO 15940
      DELGAM(N) = (GC / (WEIGHT * V0)) * ((LIFT + THRUST * SIN(ALPHA / ANG) + FC * (GA /
GC) * COS(GAMMA / ANG) - WEIGHT * (GA / GC) * COS(GAMMA / ANG))) * DT / 1
     '--------------------Altitude increment--------------------------------
15940 DELH(N) = (V0 * SIN(GAMMA / ANG)) * DT / 1
     '--------------------Range increment-----------------------------------
      DELR(N) = (V0 * COS(GAMMA / ANG)) * DT / 1
     '
     '--------------Go back for 2nd pass thru equations----------------------
     TIMEX = TIMEX + DT / 2
     WEIGHT = WEIGHT + DELW(N)
     V0 = V0 + DELV(N)
     GAMMA = GAMMA + DELGAM(N)
     H0 = H0 + DELH(N)
     RANGE = RANGE + DELR(N)
     NEXT N
      ' ------------After two passes :
      '-------------Add increments as below of two----------------------
      '              integration steps to "old" values to
      '              obtain "new" values of integration
      '               variables at "new" time
      TIMEX = TIMEX0 + DT
      WEIGHT = WEIGHT0 + (DELW(1) + DELW(2)) / 2
      V0 = V00 + (DELV(1) + DELV(2)) / 2
      M0 = V0 / C0
      GAMMA = GAMMA0 + (DELGAM(1) + DELGAM(2)) / 2
      H0 = H00 + (DELH(1) + DELH(2)) / 2
      RANGE = RANGE0 + (DELR(1) + DELR(2)) / 2
        Q0OLD = Q0: H0OLD = H0: GAMMAOLD = GAMMA: RH0OLD = RH0
      '
      '-------------------End of integration routine------------------
      RETURN
       '
      '$PAGE
```

```
16000 '**********************************************************************
      '                        TABLE LOADER SUBROUTINE
      '                 THIS SECTION PROVIDED BY KEN WATSON
      '**********************************************************************
      '
      IL = IL + 1: IF IL = 1 THEN ILTI = 0: 'Initialize table location index
      I1ST(IL) = 2: J1ST(IL) = 2
      IO(IL) = ILTI + 1: ITO = IO(IL): 'Set table origin
      INPUT #1, T$(IL), AT(ITO):      'Read table title & type
      IF AT(ITO) = 1 THEN 16090 ELSE 16150
16090 '                            1-D TABLE
      IX = ITO + 1: INPUT #1, AT(IX):  'Input number of points
      IY = IX + AT(IX)
      FOR I = 1 TO AT(IX): INPUT #1, AT(I + IX): NEXT I: 'Input x
      FOR I = 1 TO AT(IX): INPUT #1, AT(I + IY): NEXT I: 'Input y
      ILTI = AT(IX) + IY: RETURN: 'Set index of last point
16150 '                            2-D TABLE
      IX = ITO + 2: INPUT #1, AT(ITO + 1), AT(IX): 'Input N and M
      IY = IX + AT(ITO + 1): IZ = IX + AT(IX)
      FOR I = 1 TO AT(ITO + 1): INPUT #1, AT(I + IX): NEXT I: 'Input x
      FOR I = 1 TO AT(IX): INPUT #1, AT(I + IY): NEXT I: 'Input y
      FOR J = 1 TO AT(IX): FOR I = 1 TO AT(ITO + 1): 'Input z
      K = I + AT(ITO + 1) * J + IZ: INPUT #1, AT(K): NEXT I: NEXT J
      ILTI = AT(ITO + 1) * (1 + AT(IX)) + IZ: RETURN
      '
      '
      '
      '$PAGE
```

```
16230 '************************************************************************
      '                 TABLE PRINTER SUBROUTINE
      '                  PROVIDED BY KEN WATSON
      '*****************************************************************
      LPRINT : LPRINT : LPRINT "     "; T$(ITAB): ITO = IO(ITAB)
      LPRINT "     "; AT(ITO); "-D TABLE", "STARTS AT "; ITO: LPRINT
      IF AT(ITO) = 1 THEN 16290 ELSE 16340
16290 IX = ITO + 1: IY = IX + AT(IX)
      LPRINT TAB(10); "X"; TAB(25); "Y"
      FOR I = 1 TO AT(IX)
      LPRINT USING "#######.#####"; AT(I + IX), AT(I + IY): NEXT I
      RETURN
16340 IX = ITO + 2: IY = IX + AT(ITO + 1): IZ = IX + AT(ITO + 2)
      LPRINT TAB(10); "X"; TAB(25); "Y"; TAB(35); "Z"
      FOR J = 1 TO AT(ITO + 2): FOR I = 1 TO AT(ITO + 1)
      K = I + J * AT(ITO + 1) + IZ
      LPRINT USING "#######.#####"; AT(I + IX), AT(J + IY), AT(K)
      NEXT I: NEXT J: RETURN
      '
      '
      '
      '$PAGE
```

```
16400 '***********************************************************************
      '                TABLE INTERPOLATION SUBROUTINE
      '                    PROVIDED BY KEN WATSON
      '***********************************************************************
      ITO = IO(ITAB): IF AT(ITO) = 1 THEN 16440 ELSE 16590
16440 '                          1-D table
      IX = ITO + 1: N = AT(IX): IY = IX + N
      IF XARG <= AT(IX + 1) THEN 16470 ELSE 16490
16470 I = 2
16480 I2 = I + IX: I1 = I2 - 1: J2 = I + IY: J1 = J2 - 1: GOTO 16570
16490 IF XARG >= AT(IX + N) THEN 16500 ELSE 16510
16500 I = N: GOTO 16480
16510 I = I1ST(ITAB)
16520 I2 = I + IX: I1 = I2 - 1: J2 = I + IY: J1 = J2 - 1
      IF XARG <= AT(I2) THEN 16540 ELSE 16560
16540 IF XARG > AT(I1) THEN 16570 ELSE 16550
16550 I = I - 1: GOTO 16520
16560 I = I + 1: GOTO 16520
16570 VALI = AT(J1) + (AT(J2) - AT(J1)) * ((XARG - AT(I1)) / (AT(I2) - AT(I1)))
      I1ST(ITAB) = I: RETURN
      '
      '
      '
16590 '                           2-D table
      IX = ITO + 2: N = AT(ITO + 1): M = AT(ITO + 2): IY = IX + N: IZ = IX + M
      IF XARG <= AT(1 + IX) THEN 16630 ELSE 16620: 'Locate I1 & I2
16620 IF XARG >= AT(N + IX) THEN 16640 ELSE 16650
16630 I = 2: GOTO 16710
16640 I = N: GOTO 16710
16650 I = I1ST(ITAB)
16660 I2 = I + IX: I1 = I2 - 1
      IF XARG <= AT(I2) THEN 16680 ELSE 16700
16680 IF XARG > AT(I1) THEN 16710 ELSE 16690
16690 I = I - 1: GOTO 16660
16700 I = I + 1: GOTO 16660
16710 I1ST(ITAB) = I
      I2 = I + IX: I1 = I2 - 1
      IF YARG <= AT(1 + IY) THEN 16750 ELSE 16740: 'Locate J1 & J2
16740 IF YARG >= AT(M + IY) THEN 16760 ELSE 16770
16750 J = 2: GOTO 16830
16760 J = M: GOTO 16830
16770 J = J1ST(ITAB)
16780 J2 = J + IY: J1 = J2 - 1
      IF YARG <= AT(J2) THEN 16800 ELSE 16820
16800 IF YARG > AT(J1) THEN 16830 ELSE 16810
16810 J = J - 1: GOTO 16780
16820 J = J + 1: GOTO 16780
16830 J1ST(ITAB) = J
      J2 = J + IY: J1 = J2 - 1
      K11 = I - 1 + N * (J - 1) + IZ: K21 = I + N * (J - 1) + IZ
      K12 = I - 1 + N * J + IZ: K22 = I + N * J + IZ
      DXI = (XARG - AT(I1)) / (AT(I2) - AT(I1)): 'Have I1,I2,J1,J2,K11,K21,K12,K22
      DYJ = (YARG - AT(J1)) / (AT(J2) - AT(J1))
      VAL1 = AT(K11) + (AT(K21) - AT(K11)) * DXI
      VAL2 = AT(K12) + (AT(K22) - AT(K12)) * DXI
      VALI = VAL1 + (VAL2 - VAL1) * DYJ
      RETURN
      '
      '
      '$PAGE
```

```
18000 '**********************************************************************
      '                SUBROUTINE TO LOAD SECOND STAGE DATA FILE
      '**********************************************************************
          OPEN "I", #1, IFN2$: IL = 0: IFILE = 1
      '
      LINE INPUT #1, TITL1$
      LINE INPUT #1, TITL2$
      '
      GOSUB 16000:      'Input airbreather Isp vs. Mach and PHI as table #1
      GOSUB 16000:      'Input airbreather PHImax vs. Mach as table #2
      GOSUB 16000:      'Input Cdo vs.Mach as table #3
      GOSUB 16000:      'Input delta Cd vs. Alt. as table #4
      GOSUB 16000:      'Input Clalpha vs. Mach as table #5
      GOSUB 16000:      'INPUT K VS. MACH AS TABLE #6
      GOSUB 16000:      'Input airbreather Ao/Ac vs. Mach and alpha as table #7
      GOSUB 16000:      'Input rocket Isp vs. Altitude as table #8
      GOSUB 16000:      'Input rocket WDOTPMAX vs.Mach as table #9

       INPUT #1, AC, AREF
       INPUT #1, WEIGHT, WFUEL, WFINAL, VFINAL, STAGE
       INPUT #1, DT, DELPRINT
       INPUT #1, Q0COMD, Q0MAX, Q0MIN, ALPHAMAX, ALPHAMIN
      '
       INPUT #1, LOADFAC, ACCCOMD, GAMMAMAX
       INPUT #1, FASTOIC
       INPUT #1, V0TEMP, Q0FINAL
       INPUT #1, V0CRUISE, SWITCH4, GAMMA4MAX
       INPUT #1, FLAG5, V05, ALPHA5MAX, GAMMA5, ACCCOMD5
       INPUT #1, A1, A2, B1, C1, C2, D1, DELQ, V02, H02
       '
       PRINT
       PRINT "INPUT FILE FOR 2ND STAGE HAS BEEN READ."
       PRINT "LAST ITEM IN 2ND STAGE INPUT FILE IS H02,VALUE READ FROM FILE IS"; H02
       PRINT "IF THIS VALUE IS INCORRECT THEN INPUT FILE IS IN ERROR."
       PRINT
      INPUT "Ready to run 2nd stage:
   Do you want a listing of 2nd stage tabular input data(Y/N)"; WTIN$
      IF WTIN$ = "Y" OR WTIN$ = "y" THEN 18230 ELSE 18240
18230 FOR ITAB = 1 TO 9: GOSUB 16230: NEXT ITAB
18240     TIMEX2 = TIMEX
          IF EOF(1) THEN GOTO 18250
18250     CLOSE #1
      RETURN:              'Returns to main routine
      '
      '
      '$PAGE
```

```
20000 'PRIN.BAS ******************************************************************
      '                  POST-RUN PRINTER OUTPUT SUBROUTINE
      '*******************************************************************************
      ' 8MAY87/0030/PROGRAM "ETOTAB.BAS",MERGED FOR USE WITH PROGRAM "ETO"
      N = 55:  'Number of lines printed per page
      WIDTH LPRINT 136
      '
      LPRINT TAB(40); DATE$; TAB(60); TIME$
      LPRINT
      LPRINT TAB(10); "1ST/SINGLE STAGE INPUT DATA FILE: "; IFN$
      LPRINT TAB(10); "2ND STAGE INPUT DATA FILE       : "; IFN2$
      LPRINT
      LPRINT TAB(10); "PLOTTER/PRINTER FILE NAME       : "; PFN$
      LPRINT
      LPRINT
      LPRINT TAB(10); "------PROGRAM ETO2 RUN SUMMARY--------"
      LPRINT USING "              LAUNCH WEIGHT = #########.## LBS."; WLAUNCH
      LPRINT USING "               FINAL WEIGHT  = #########.## LBS."; WEIGHT
      LPRINT USING "FUEL/PROPELLANT REMAINING =#########.## LBS."; WEIGHT - WFINAL
      LPRINT
      IF V0 >= VORBIT THEN LPRINT TAB(10); "VEHICLE REACHED ORBITAL VELOCITY."
      IF WEIGHT <= WFINAL THEN LPRINT TAB(10); "RUN TERMINATED BY FUEL EXHAUSTION"
      LPRINT
      LPRINT USING "FLIGHT TIME    = #######.## SECS."; TIMEX
      LPRINT USING "FINAL VELOCITY = #######.## FT/SEC"; V0
      LPRINT USING "FINAL ALTITUDE = #######.## FT"; H0
      LPRINT USING "FINAL Q0       = #######.## PSF"; Q0
      LPRINT
      LPRINT USING "AVERAGE ISP    = #######.## SEC "; ISPAVG
      LPRINT USING "MEAN ISP       = #######.## SEC "; ISPMEAN
      '
      LPRINT
      LPRINT
      LPRINT
      LPRINT "FINAL CONDITIONS ARE SHOWN BELOW:"
      LPRINT
      WIDTH LPRINT 136
      ''' LPRINT CHR$(27); "B"; CHR$(3):'Selects condensed print(SR15 printer)
      LPRINT CHR$(27); CHR$(33); CHR$(4); : 'Selects condensed print(Epson printer)
      LPRINT TAB(5); "TIME"; TAB(15); "V0"; TAB(25); "H0"; TAB(35); "R"; TAB(45); "ISP";
TAB(55); "ISPEFF"; TAB(65); "ISPMEAN"; TAB(75); "WDOTF"; TAB(85); "WDOTA"; TAB(97); "PHI"
; TAB(105); "THRUST"; TAB(112); "THRUSTCOM"; TAB(122); "THRUSTMAX"
      LPRINT TAB(15); "Q0"; TAB(25); "ALPHA"; TAB(35); "GAMMA"; TAB(45); "WEIGHT"; TAB(55
); "LIFT"; TAB(65); "DRAG"; TAB(75); "WDOTP"; TAB(85); "RES1"; TAB(94); "ISPAVG"; TAB(102
); "ACCEL-G'S"; TAB(115); "ACCCOM"; TAB(128); "T/D"
      LPRINT
      LPRINT USING "#######.##"; TIMEX; V0; H0; RANGE / 6076; ISP; ISPEFF; ISPMEAN;
WDOTF; WDOTA; PHI; THRUST; THRUSTCOM; THRUSTMAX
      LPRINT USING "#######.##"; TIMEX; Q0; ALPHA; GAMMA; WEIGHT; LIFT; DRAG; WDOTP;
RES1; ISPAVG; ACC; ACCCOM; THRUST / DRAG
      WIDTH LPRINT 80
      LPRINT CHR$(12): 'Form feed
      '''LPRINT CHR$(27) "B" CHR$( 1) :'Resets to standard print(SR15 printer)
      LPRINT CHR$(27); CHR$(33); CHR$(0); : 'Resets to standard print(Epson printer)
      '
      CMD1$ = "COPY" + CHR$(32) + IFN$ + CHR$(32) + "PRN"
      PRINT CMD1$
      SHELL CMD1$
      LPRINT CHR$(12): 'Form feed
      IF NSTAGES = 1 THEN GOTO 20820
      CMD2$ = "COPY" + CHR$(32) + IFN2$ + CHR$(32) + "PRN"
      PRINT CMD2$
      SHELL CMD2$
      LPRINT CHR$(12)
20820 WIDTH LPRINT 136
      '''LPRINT CHR$(27) "B" CHR$(3) :'Selects condensed print(SR15 printer)
      LPRINT CHR$(27); CHR$(33); CHR$(4); : 'Selects condensed print(Epson printer)
      '
      '
      DIM V(25)
      COUNT = 0
```

```
      LPRINT TAB(10); PFN$; TAB(40); DATE$; TAB(60); TIME$
      GOSUB 22460
      '
      OPEN "I", 2, PFN$: INPUT #2, VARNUM
21140 FOR I = 1 TO 25
          INPUT #2, V(I)
              IF EOF(2) THEN GOTO 21340
       NEXT I
      COUNT = COUNT + 1
      LPRINT USING "#########.##"; V(1); V(3); V(2); V(14); V(4); V(17); V(8); V(9); V(10); V(20); V(21)
      IF COUNT >= N THEN LPRINT CHR$(12)
      IF COUNT >= N THEN COUNT = 0
      IF COUNT = 0 THEN GOSUB 22460
      GOTO 21140
21340 CLOSE 2
      '''''''''
      GOTO 21700: 'SKIPS OVER 2ND FIELD FOR PROGRAM "ETO"
       LPRINT CHR$(12): 'FORM FEED
       COUNT = 0
      GOSUB 22500
      OPEN "I", 2, PFN$: INPUT #2, VARNUM
21480 FOR I = 1 TO 25
          INPUT #2, V(I)
              IF EOF(2) THEN GOTO 21680
       NEXT I
      COUNT = COUNT + 1
      LPRINT USING "#########.##"; V(1); V(3); V(2); V(8); V(9); V(10); V(29); V(28); V(20)
      IF COUNT >= N THEN LPRINT CHR$(12)
      IF COUNT >= N THEN COUNT = 0
      IF COUNT = 0 THEN GOSUB 22500
      GOTO 21480
21680 CLOSE 2
21700 ''''''
       LPRINT CHR$(12): 'FORM FEED
      COUNT = 0
      GOSUB 22540
      OPEN "I", 2, PFN$: INPUT #2, VARNUM
21800 FOR I = 1 TO 25
          INPUT #2, V(I)
              IF EOF(2) THEN GOTO 22000
       NEXT I
      COUNT = COUNT + 1
      LPRINT USING "#########.##"; V(1); V(3); V(2); V(5); V(22); V(6); V(7); V(11); V(12); V(13)
      IF COUNT >= N THEN LPRINT CHR$(12)
      IF COUNT >= N THEN COUNT = 0
      IF COUNT = 0 THEN GOSUB 22540
      GOTO 21800
22000 CLOSE 2
      ''''''
       LPRINT CHR$(12): 'FORM FEED
      COUNT = 0
      GOSUB 22580
      OPEN "I", 2, PFN$: INPUT #2, VARNUM
22120 FOR I = 1 TO 25
          INPUT #2, V(I)
              IF EOF(2) THEN GOTO 22320
       NEXT I
      COUNT = COUNT + 1
      LPRINT USING "#########.##"; V(1); V(3); V(2); V(14); V(15); V(16); V(18); V(19); V(23); V(24); V(25)
      IF COUNT >= N THEN LPRINT CHR$(12)
      IF COUNT >= N THEN COUNT = 0
      IF COUNT = 0 THEN GOSUB 22580
      GOTO 22120
22320 CLOSE 2
      ''''''
      ''''''''
      ''' LPRINT CHR$(27); "B"; CHR$(1):       'Resets to standard print
      LPRINT CHR$(27); CHR$(33); CHR$(0); : ' Resets to standard print(Epson printer)
```

```
      WIDTH LPRINT 80
      LPRINT CHR$(12):     'Form feed
      GOTO 22999
22460 LPRINT TAB(6); "TIME"; TAB(18); "H0"; TAB(30); "V0"; TAB(42); "Q0"; TAB(54); "R";
TAB(66); "WEIGHT"; TAB(78); "WDOTF  "; TAB(90); "  WDOTA   "; TAB(102); " PHI  "; TAB(114)
; "   WDOTP "; TAB(125); "  RES1 "
      RETURN
22500 LPRINT TAB(6); "TIME"; TAB(18); "H0"; TAB(30); "V0"; TAB(42); "WDOTF"; TAB(54); "WD
OTA"; TAB(66); "PHI"; TAB(78); "WDOTFROC"; TAB(90); "WDOTLOX"; TAB(102); "WDOTP"
      RETURN
22540 LPRINT TAB(6); "TIME"; TAB(18); "H0"; TAB(30); "V0"; TAB(42); "ISP"; TAB(54); "ISPA
VG"; TAB(66); "ISPEFF"; TAB(78); "ISPMEAN"; TAB(90); "THRUST"; TAB(102); "THRUSTCOM";
TAB(114); "THRUSTMAX"
      RETURN
22580 LPRINT TAB(6); "TIME"; TAB(18); "H0"; TAB(30); "V0"; TAB(42); "Q0"; TAB(54); "ALPHA
"; TAB(66); "GAMMA"; TAB(78); "LIFT"; TAB(90); "DRAG"; TAB(102); "ACCEL-G'S"; TAB(114); "
ACCCOM"; TAB(129); "T/D"
      RETURN
22999 RETURN
```

5.0 CROSS REFERENCED LISTING OF PROGRAM ETO VARIABLES

Each variable used in the program is listed alphabetically along with the sequential program line numbers for each occurrence. The line numbers are consistent with the program listing in Section 4.0.

```
         A$  192 192 226 391
         A0  558 559 560
       A0AC  180 553 558 598
         AI   91 268 827
         A2   91 268 827
         AC   81 558 817
        ACC  378 399 422 633 887
     ACCCOM  221 240 241 294 303 305 322 357 398 422 550 633
             887
    ACCCOMD   86 221 240 294 302 303 822
   ACCCOMD5   90 357 826
    ACCPERQ  302 303
      ALPHA   83 180 184 220 223 238 239 239 242 268 268 273
             273 278 278 283 283 285 285 286 286 326 331 333
             336 339 342 343 344 345 358 359 393 422 542 550
             550 553 633 654 657 887
  ALPHA2MAX   85 239 239
  ALPHA5MAX   90 358 358 826
ALPHACRUISE  339 342
   ALPHAMAX   84 285 285 326 345 358 396 820
   ALPHAMIN   84 286 286 397 820
        ANG  143 550 550 550 654 654 657 657 657 659 661
       AREF   81 185 187 238 543 545 817
         AT   41 697 698 700 701 702 702 703 703 704 706 706
             707 707 708 708 709 709 710 710 711 711 712 712
             722 723 724 726 727 727 729 729 731 731 732 733
             733 733 743 745 746 749 753 754 757 757 757 757
             757 757 763 763 764 765 770 771 776 777 782 783
             790 790 790 791 791 791 792 792 792 793 793 793
         B$  244 391
         BI   91 273 827
       BIGG  141 203
         C$  315 391
         C0  525 526 679
         CI   91 278 827
         C2   91 278 827
         CD  186 187 332 333 544 545
        CD0  176 186 327 332 538 544
         CL  184 185 186 331 332 333 542 543 544
    CLALPHA  178 184 238 329 331 540 542
      CMD1$  893 894 895
      CMD2$  898 899 900
      COUNT  149 388 388 389 404 908 917 917 919 920 920 921
             927 934 934 936 937 937 938 943 950 950 952 953
             953 954 959 966 966 968 969 969 970
       CPCV  504
         D$  348 349 391
         DI   91 283 827
      DELCD  177 186 328 332 539 544
     DELGAM   43 657 667 680 680
       DELH   43 659 668 681 681
   DELPRINT   83 389 819
       DELQ   91 271 272 281 282 827
       DELR   43 661 669 682 682
  DELTHRUST  586 587
       DELV   43 654 666 678 678
       DELW   43 651 665 677 677
       DRAG  187 398 399 400 422 422 545 550 633 633 654 887
             887
         DT   83 205 208 209 209 226 235 243 267 270 271 277
             280 281 378 651 653 654 656 657 659 661 664 676
             819
         DV  207 208
        DXI  790 792 793
        DYJ  791 794
          E  139 140 141 503
         E$  363 391
      FAMAX  561 563
    FASTOIC   87 561 607 613 823
         FC  142 397 528 657
     FINISH  458 459
      FLAG5   90 261 314 826
```

```
FLAGCRUISE  324  341
      FREQ  119  631
     FREQC  631  632  634  634
        G0  529  560
        GA   42  203  378  550  550  654  657  657
     GAMMA   83  220  223  343  345  361  394  422  550  633  642  654
            657  657  659  661  667  667  680  683  887
    GAMMA0  642  680
 GAMMA4MAX   89  343  825
    GAMMA5   90  361  826
  GAMMAMAX   86  822
  GAMMAOLD  683
        GC  138  142  203  372  382  502  516  520  522  527  528  550
            654  654  657  657  657
      GLMB   44  155  515  519  520
         H  506  509  516  519
        H0   83  142  177  203  220  223  237  241  242  312  328  382
            394  421  506  506  528  539  567  633  643  668  668  681
            683  869  886
       H00  643  681
       H02   91   95  237  241  242  827  831
     H0OLD  312  683
        HB   44  155  509  516  519
      HDEL  509  510
         I  151  190  427  702  702  702  703  703  703  708  708  708
            709  709  709  710  711  711  726  727  727  727  731  732
            733  734  747  748  748  750  751  752  752  755  755  756
            756  758  766  767  768  769  772  772  773  773  774  775
            788  788  789  789  913  914  916  930  931  933  946  947
            949  962  963  965
        I1  508  509  511  515  516  516  516  517  519  519  519  520
            520  520  748  752  754  757  757  769  771  775  790  790
      I1ST   41  695  751  758  768  774
        I2  748  748  752  752  753  757  769  769  770  775  775  790
     IFILE   52  802
      IFN$   47   49   52  853  893
     IFN2$   50  802  854  898
        IL   52  694  694  694  695  695  696  696  697  802
      ILTI  694  696  704  712
        IO   41  696  696  721  743
       ISP  376  396  421  576  595  616  633  886
      ISPA  400  556  564  598  602  604  614  616  618
    ISPAVG  377  399  422  633  872  887
    ISPEFF  372  373  397  421  633  886
   ISPMEAN  374  398  421  633  873  886
      ISPR  401  567  570  591  619
      ITAB   70   70  176  177  178  179  180  327  328  329  330  538
            539  540  541  553  554  556  567  568  602  721  721  743
            751  758  768  774  780  786  836  836
     ITCUM  373  373  374
       ITO  696  697  698  700  706  706  707  708  710  711  712  721
            722  722  723  724  729  729  729  731  731  732  743  743
            745  763  763  763
     ITSUM  376  376  377
        IX  700  700  701  701  702  702  703  704  706  706  707  707
            707  708  709  710  712  724  724  724  726  727  729  729
            729  733  745  745  745  746  748  749  752  763  763  763
            764  765  769  775
        IY  701  703  704  707  709  724  727  729  733  745  748  752
            763  776  777  781  787
        IZ  707  711  712  729  732  763  788  788  789  789
         J  710  711  711  731  732  733  734  778  779  780  781  784
            784  785  785  786  787  788  788  789  789
        J1  153  155  155  155  155  156  748  752  757  757  781  783
            787  791  791
      J1ST   41  695  780  786
        J2  748  748  752  752  757  781  781  782  787  787  791
         K  179  186  330  332  541  544  711  711  732  733
       K11  788  792  792
       K12  789  793  793
       K21  788  792
       K22  789  793
```

```
        LIFT  185  220  223  344  396  398  398  422  543  633  657  887
     LOADFAC   86  238  822
       LOVRD  333  333  334  335
    LOVRDOLD  334  335
           M  763  763  777  779
          M0   83  176  178  179  180  327  329  330  393  526  538  540
              541  553  554  556  568  602  679
      MEARTH  139  203
           N  647  651  654  657  659  661  665  666  667  668  669  670
              745  745  749  750  763  763  765  767  788  788  789  789
              848  919  920  936  937  952  953  968  969
          N1  501  508
    NSTAGES    45   46  443  444  897
          P0  523  529
        PFN$  100  101  856  909  912  929  945  961
       PHASE  216  230  252  321  354  391  391  391  391  391
         PHI  395  421  555  556  559  561  599  602  607  613  633  886
      PHIMAX  554  555  598  599
       PLOT$   98   99   99  446  446  451  451  452  453  453  455  455
              629  629
       PRCOM   73   73   74  407  434  437  440  446
      PRCOM$   72   73   73
     PRCOUNT  150  409  410  412  418  418  425  425
          Q$  194  194  197  198
          Q0  185  187  237  238  264  265  267  270  271  271  272  275
              277  280  281  281  282  284  291  303  305  310  395  422
              527  543  545  633  683  870  887
       Q0COM  259  262  264  270  271  272  280  281  282  291  302  303
      Q0COMD   84  237  259  262  262  820
     Q0FINAL   88  262  824
       Q0MAX   84  302  305  820
       Q0MIN   84  820
       Q0OLD  310  683
        QOLD  265  267  271  275  277  281  284
          R0  516  520  522
       RANGE  395  421  633  644  669  669  682  886
      RANGE0  644  682
    RATEDELQ  270  273  280  283
      RATEDQ  267  268  271  277  278  281
          RE  503  506  506
      REARTH  140  142  203  382  382  528
        RES1  393  422  633  887
         RH0  311  522  523  527  683
        RH0B   44  155  516  520
      RH0OLD  311  683
       STAGE   82  385  444  818
       START  120  459
  STARTTIME$  120  458
     SWITCH4   89  323  349  825
          T$   41  697  721
          T0  517  519  520  523  525  529
          TG   42
      THRUST  400  400  402  421  422  572  576  578  594  595  597  597
              604  605  609  614  633  633  654  657  886  887
  THRUSTAMAX  564  571  590  594
   THRUSTCOM  401  421  550  578  586  590  605  609  633  886
   THRUSTMAX  402  421  571  572  586  633  886
     THRUSTR  590  591  594
  THRUSTRMAX  570  571  587
       TIMEX   83  148  218  235  392  421  422  433  434  436  437  439
              440  445  633  639  653  664  664  676  837  867  886  887
      TIMEX0  639  676
      TIMEX2  445  837
      TITL1$   55  412  804
      TITL2$   56  412  805
         TMB   44  155  516  517  519  520
         TQ0  235  243  243
       TSET5  261  359  360  361  361
    TTAKEOFF  218  226  226  235
           V  907  914  918  918  918  918  918  918  918  918  918  918
              918  931  935  935  935  935  935  935  935  935  935  947
              951  951  951  951  951  951  951  951  951  951  963  967
```

```
             967  967  967  967  967  967  967  967  967  967
        V0   142  206  207  218  237  241  260  262  313  314  372  378
             383  385  393  421  433  434  436  437  439  440  526  527
             528  529  633  641  656  657  659  661  666  666  678  679
             864  868  886
       V00   372  378  641  678
       V02    91  237  241  827
       V05    90  314  826
  V0CRUISE    89  313  313  825
    V0TEMP    88  260  262  262  824
      VAL1   792  794  794
      VAL2   793  794
      VALI   176  177  178  179  180  327  328  329  330  538  539  540
             541  553  554  556  567  568  602  757  794
    VARNUM   102  103  912  929  945  961
    VFINAL    82  385  818
    VORBIT   382  383  864
  VTAKEOFF    85  218
      WDOT   575  576  593  595  615  651
     WDOTA   394  421  560  563  607  613  633  886
     WDOTF   393  421  573  592  593  604  605  605  607  613  614  615
             618  633  886
  WDOTFMAX   563  564  573  575  592
     WDOTP   394  422  574  587  588  591  593  619  633  887
  WDOTPMAX   568  570  574  575  584
    WEIGHT   142  148  205  220  223  238  344  372  372  373  374  376
             377  384  396  396  402  422  433  434  436  437  439  440
             528  550  633  640  653  654  654  657  657  665  665  677
             818  861  862  865  887
   WEIGHT0   372  373  376  640  677
    WFINAL    82  205  205  384  818  862  865
     WFUEL    82  818
   WLAUNCH    82  148  374  377  653  860
     WTIN$    68   69   69  834  835  835
        X$   391  391  391  391  391  392
      XARG   176  177  178  179  180  327  328  329  330  538  539  540
             541  553  554  556  567  568  602  746  749  753  754  757
             764  765  770  771  790
      YARG   180  553  556  602  776  777  782  783  791
```

6.0 DEFINITION OF PROGRAM ETO VARIABLES

The program variables in this section are listed alphabetically by the following groupings:

- Simple Real Variables
- Simple String Variables
- Array Real Variables
- Array String Variables

VARIABLE LISTING FOR PROGRAM "ETO"

Simple Real Variables

A0	Air capture streamtube,sq.ft.	Internal
A0AC	Air capture/cowl area ratio.Values are input as table 6 of input data file.	Input
A1	Adjustment "knob" for controlling Phase 3 constant q path.Presently set to 0.1	Input
A2	Adjustment "knob" for controlling Phase 3 constant q path.Presently set to 0.1	Input
AC	Air inlet cowl area,sq.ft.	Input
ACC	Axial acceleration,g's.Output is labeled ACCEL-G'S.	Output
ACCCOM	Commanded axial acceleration,g's	Output
ACCCOMD	Maximum axial acceleration,g's	Input
ACCCOMD5	Commanded acceleration in Phase 5,g's	Input
ACCPERQ	Variable used to adjust acceleration in Phase 3,constant q path.	Internal
ALPHA	Angle of attack,degrees	Output
ALPHA2MAX	Maximum angle of attack in Phase 2,degrees	Input
ALPHA5MAX	Maximum angle of attack in Phase 5,degree	Input
ALPHACRUISE	Angle of attack for maximum L/D cruise in Phase 4	Internal
ALPHAMAX	Maximum angle of attack in Phase 1 thru 4,degrees.	Input
ALPHAMIN	Minimum angle of attack,degrees	Input
ANG	57.3 degrees/radian	Internal
AREF	Vehicle lift and drag reference area,sq.ft.	Input
B1	Adjustment "knob" for controlling Phase 3 constant q path.Presently set to 0.009	Input
BIGG	Universal gravitational constant	Internal
C0	Velocity of sound,ft./sec.	Internal
C1	Adjustment "knob" for controlling Phase 3 constant q path. Presently set to 0.1	Input
C2	Adjustment "knob" for controlling Phase 3 constant q path.Presently set to 2.0	Input
CD	Vehicle drag coefficient	Internal
CD0	Zero lift drag coefficient.Values are input as table 3 of input data file.	Input
CL	Lift coefficient	Internal
CLALPHA	Lift coefficient/degree.Values are input as table 5 of input data file.	Input
COUNT	Loop counter for controlling screen and printer interval.	Internal
CPCV	Air specific heat ratio.Not used	Internal
D1	Adjustment "knob" for controlling Phase 3	Input

	constant q path.Presently set to 0.1	
DELCD	Friction increment of drag.Values are input as table 4 of input data file.	Input
DELPRINT	Print and screen output interval	Input
DELQ	Adjustment "knob" for controlling Phase 3 constant q path.A value equal to 5% of Q0COMD is suggested.	Input
DELTHRUST	Iteration tolerance on commanded thrust	Internal
DRAG	Vehicle drag,lbf.	Output
DT	Trajectory integration interval,sec.	Input
DV	Reduced integration interval used near end of flight,secs.	Internal
DXI	Variable in table interpolation routine.	Internal
DYJ	Variable in table interpolation routine.	Internal
FAMAX	Maximum fuel/air ratio	Internal
FASTOIC	Stoichiometric fuel/air ratio	Input
FC	Vehicle centrifugal force,lbf.	Internal
FINISH	Clock time at end of problem,hr/min/sec	Output
FLAG5	Selector for Phase 5,ballistic ascent. Enter 0 if not used,1 if ballistic phase is used.	Input
FLAGCRUISE	If vehicle is to cruise at maximum L/D, this variable is reset from zero to a value of 1 to bypass optimum alpha search once it is made.	Internal
FREQ	Frequency of output to plotter file	Input at prompt
FREQC	Plotter file counter variable	Internal
G0	Freestream specific airflow,lbm/(sec-sq in)	Internal
GA	Gravitational acceleration,ft/sec-sec	Internal
GAMMA	Vehicle flight path angle,degrees	Internal
GAMMA0	"Old" flight path angle at start of integration interval,degrees	Internal
GAMMA4MAX	Maximum flight path angle permitted during Phase 4 maximum L/D cruise,degrees. A small value,less than 0.5 degrees is suggested.	Input
GAMMA5	Flight path angle to be achieved before ballistic path begins,degrees	Input
GAMMAMAX	Maximum allowed flight path angle for all phases except 4 and 5,degrees.	Input
GAMMAOLD	Same as GAMMA0,degrees	Internal
GC	32.174 lbm ft per sec sq/lbf	Internal
H	Geopotential altitude,ft.	Internal
H0	Vehicle flight altitude,ft.An initial value is input.Values during run are output.	Input and Output
H00	"Old" value of flight altitude at start of integration interval,ft.	Internal

H02	Phase 2 altitude to be achieved before V02 can be exceeded,ft.	Input
H0OLD	Same as H00	Internal
HDEL	Altitude variable used in atmosphere routine,ft.	Internal
I	Integration loop counter and index variable	Internal
I1	Counter and index variable	Internal
I2	Counter and index variable	Internal
IFILE	Input file designator	Internal
IL	Input file designator	Internal
ILTI	Index variable in table loader	Internal
ISP	Specific impulse,secs.Sum of airbreathing and rocket thrust divided by sum of air-breathing fuel and rocket propellant flows.	Output
ISPA	Specific impulse of airbreather,sec. Values are input as table 1 of input data file.	Input and Output
ISPAVG	Running average specific impulse since start of flight,secs.	Output
ISPEFF	Instantaneous effective specific impulse defined by (weight/gc)X(delta velocity/delta weight.	Output
ISPMEAN	Running mean value of ISPEFF since start of flight.	Output
ISPR	Rocket specific impulse,secs.Values are input as table 7 of input data file.	Input and Output
ITAB	Input table number.Eight tables are currently used in input data file.	Input
ITCUM	Cumulative total effective impulse since start of flight,lb.secs.	Internal
ITO	Input data table variable.	Internal
ITSUM	Cumulative total propulsive impulse since start of flight,lb.secs.	Internal
IX	Variable used in table loader and lookup.	Internal
IY	Variable used in table loader and lookup.	Internal
IZ	Variable used in table loader and lookup.	Internal
J	Variable used in table loader and lookup.	Internal
J1	Variable used in table loader and lookup.	Internal
J2	Variable used in table loader and lookup.	Internal
K	Ratio of specific heats.Also a variable used in table loader and lookup.	Internal
K11	Variable used in table interpolation.	Internal
K12	Variable used in table interpolation.	Internal
K21	Variable used in table interpolation.	Internal
K22	Variable used in table interpolation	Internal
LIFT	Vehicle lift,lbf.	Output

LOADFAC	Maximum vertical acceleration allowed during Phase 2 climb,g's.	Input
LOVRD	L/D value during Phase 4 cruise alpha search	Internal
LOVRDOLD	"Old" L/D value used in Phase 4 cruise alpha search	Internal
M	Variable used in table interpolation	Internal
MØ	Flight Mach number.Input a small starting value,say .ØØ1.	Input and Output
MØØ	"Old" value of Mach number	Internal
MEARTH	Mass of the earth	Internal
N	Integration counter for equations of motion	Internal
N1	Variable in atmosphere routine	Internal
NSTAGES	Number of vehicle stages,1 or 2.	Input at prompt
PØ	Atmospheric static pressure,psia.	Internal
PHASE	Flight phase designator.Values are 1 thru 5.	Internal
PHI	Airbreathing fuel equivalence ratio.	Output
PHIMAX	Maximum airbreathing fuel equivalence ratio.Values are input as table 2 of input data file.	Input
PRCOM	Command for printer output during run. It is set to 1 by responding "Y" at prompt for printer output decision.	Internal
PRCOUNT	Counter for desnignating number of output points to be printed per page.	Internal
QØ	Flight dynamic pressure,lbf/sq.ft.	Output
QØCOM	Calculated dynamic pressure command	Internal
QØCOMD	Commanded dynamic pressure at start of Phase 3,lbf/sq.ft.	Input
QØFINAL	Desired fimal dynamic pressure at orbital or final condition,lbf/sq.ft.	Input
QØMAX	Maximum dynamic pressure allowed during Phase 3,lbf/sq.ft.	Input
QØMIN	Minimum dynamic pressure allowed during Phase 3,lbf/sq.ft.	Input
QØOLD	"Old" value of dynamic pressure,lbf/sq.ft.	Internal
QOLD1	Same as QØOLD.	Internal
RØ	Density variable in atmosphere routine.	Internal
RANGE	Flight range,nautical miles.	Output
RANGEØ	"Old" value of range,nm.	Internal
RATEDELQ	Commanded rate of change of dynamic pressure in Phase 3,psf/sec.	Internal
RATEDQ	Current rate of change of dynamic pressure,psf/sec.	Internal
RE	Earth radius	Internal
REARTH	Same as RE	Internal

RES1	Not used.	
RHØ	Atmospheric density,lbm./cu.ft.	Internal
RHØB	Density variable used in atmosphere routine	Internal
RHØOLD	"Old" value of density,lbm./cu.ft.	Internal
STAGE	Identifier in data table for first stage or second stage.Enter 1 or 2 in input data file.	Input
START	Clock time at start of problem,hr/min/sec	Output
SWITCH4	Used for Phase 4 cruise.Zero selects constant altitude cruise.1 selects constant L/D cruise.	Input
TØ	Static temperature of atmosphere,degrees-R.	Internal
THRUST	Vehicle thrust,lbf.	Output
THRUSTAMAX	Maximum available airbreathing thrust,lbf.	Internal
THRUSTCOM	Commanded thrust,lbf.	Output
THRUSTMAX	Maximum total thrust available,lbf.	Output
THRUSTR	Rocket thrust,lbf.	Internal
THRUSTRMAX	Maximum rocket thrust available,lbf.	Internal
TIMEX	Flight time since takeoff,secs. Input small initial value,say.ØØ1.	Output and Input
TIMEXØ	"Old" value of time,secs.	Internal
TIMEX2	Flight time at start of 2nd stage,secs.	Internal
TQØ	Time variable for testing if Phase 2 is completed.	Internal
TSET5	Time variable in Phase 5 to indicate that commanded flight path angle has been reached.When value becomes 1,ballistic ascent can begin.	Internal
TTAKEOFF	Time variable used to test if takeoff is completed.	Internal
V	Identifier for printer output variables	Internal
VØ	Vehicle flight velocity,ft./sec.	Output
VØØ	"Old" value of flight velocity,ft./sec.	Internal
VØ2	Maximum velocity permitted in Phase 2 until HØ2 altitude is reached,ft./sec.	Input
VØ5	Velocity at which Phase 5 pullup to ballistic ascent is to begin,ft./sec.	Input
VØCRUISE	Commanded Phase 4 cruise velocity,ft./sec.	Input
VØTEMP	Velocity at which commanded dynamic pressure is made a function of velocity,ft./sec. This variable is used with QØFINAL to approximate a temperature limited ascent path.	Input
VAL1	Variable in table interpolation.	Internal
VAL2	Variable in table interpolation.	Internal
VALI	Variable returned from table interpolation	Internal
VARNUM	Number of output variables written to plotter file.	Internal

VFINAL	Commanded final flight velocity,ft./sec. For a single stage going to orbit,enter a value beyond orbital velocity,say 27000. For a single stage suborbital vehicle, enter the desired final velocity. For a 2-stage vehicle,enter the desired staging velocity in the 1st stage data table and enter 27000 or a desired sub-orbital velocity in the 2nd stage data table.	Input
VORBIT	Orbital velocity,ft./sec.	Output
VTAKEOFF	Commanded takeoff velocity,ft./sec.	Input
WDOT	Total airbreathing fuel and rocket propellant flow rate,lbm./sec.	Internal
WDOTA	Propulsive air flow,lbm./sec.	Output
WDOTF	Airbreather fuel flow,lbm./sec.	Output
WDOTFMAX	Maximum airbreather fuel flow,lbm./sec.	Internal
WDOTP	Rocket propellant flow rate,lbm./sec.	Output
WDOTPMAX	Maximum rocket propellant flow rate,lbm./sec. Values are input as table 8 of input data file.	Input
WEIGHT	Vehicle weight,lbm.	Output
WEIGHT0	"Old" value of vehicle weight,lbm.	Internal
WFINAL	Vehicle final weight,lbm.	Input
WFUEL	Vehicle fuel+propellant weight,lbm. Value not required,enter zero if desired.	Input
WLAUNCH	Starting weight of vehicle,lbm.	Input
XARG	Table lookup variable	Internal
YARG	Table lookup variable	Internal

Simple String Variables

A$	Phase1,takeoff roll	Output
B$	Phase 2,climb at constant L/W	Output
C$	Phase 3,climb at commanded Q	Output
CMD1$	Command string to print 1st stage input file to printer.	Internal
CMD2$	Command string to print 2nd stage input file to printer.	Internal
D$	Phase 4,cruise at V0cruise	Output
E$	Phase 5,pullup and ballistic ascent	Output
IFN$	Input data file name (single stage or two stage) read from screen at prompt.	Input at prompt
IFN2$	Input data file name for 2nd stage.Name is read from screen at prompt.	Input at prompt
PFN$	Plotter file name entered at prompt	Input at prompt
PLOT$	Set to "Y" or "N" in response to prompt for decision to create plotter file.	Input at prompt
PRCOM$	Set to "Y" or "N" in response to prompt	Input at

	for decision for printer output during run.	prompt
Q$	Used to halt/resume program	Internal
STARTTIME$	Clock time at start of program,hr/min/sec	Output
TITL1$	Text of first line of input data file.	Input
TITL2$	Text of second line of input data file.	Input
WTIN$	Set to "Y" or "N" in response to prompt for listing of tabular data in input file.	Input at prompt
X$	Variable which is set to A$ thru E$ as each flight Phase(1 thru 5) is reached.	Internal

Array Real Variables

AT	Sets number of points in tables.	Internal
DELGAM	Change in flight path in one integration pass, degrees.	Internal
DELH	Change in altitude in one integration pass,feet.	Internal
DELR	Change in range in one integration pass, feet.	Internal
DELV	Change in velocity in one integration pass,ft./sec.	Internal
DELW	Change in weight in one integration pass, lbm.	Internal
GA	Not used as array variable.	Internal
GLMB	Variable used in atmosphere routine.	Internal
HB	Variable used in atmosphere routine.	Internal
I1ST	Variable used in table loader.	Internal
IO	Variable used in table loader.	Internal
J1ST	Variable used in table loader.	Internal
RHØB	Air density variable used in atmosphere routine.	Internal
TG	Not used.	Internal
TMB	Temperature variable used in atmosphere routine.	Internal

Array String Variables

T$	Table title used in table loader.	Internal

©U.S.Government Printing Office: 1989 - 748-056/24148